EXHIBIT 1
TO THE COMMENTS
OF RELPROMAX ANTITRUST INC.

ECONOMIC MODEL FOR ANALYSIS OF COMPETITIVE IMPACT OF THE RPFJ

A. THE MODELING PROBLEMS CAUSED BY THE SHORTCOMINGS OF THE COMPETITIVE IMPACT STATEMENT

As indicated above, the Competitive Impact Statement provides no economic analysis or economic modeling of any kind. The Competitive Impact Statement does not even provide raw economic data upon which an economic analysis might be made. It provides no information concerning revenues, costs, profits, quantities, or product qualities of Microsoft, its competitors, or potential competitors which might usefully be incorporated into an economic model. The CIS does not indicate the United States reviewed or considered any such items (i.e. revenues, costs, profits, quantities, or product qualities of Microsoft, its competitors, or potential competitors) in connection with the RPFJ or the CIS. The DOJ's "Competitive Impact Statement" may be a "statement" of sorts, but it is clearly not a statement of "competitive impacts," about which the statement truly says nothing at all.

This places a heavy burden on the public. Members of the public who wish to critique the consent agreement, must not only devise their own economic models and collect their own economic data, they can only guess at what economic models and economic analysis the DOJ is hiding from the public.

Accordingly, a member of the public who wishes to comment is forced to devise her own economic models and collect her own economic data. In the case of this model, the work has been performed by a professional economist. It would be preferable to use or critique the DOJ's own economic models of the software industry. However, the DOJ has provided no such economic models and no analysis of the competitive impact of the Revised Proposed Final Judgment.[1]

B. How an Economist Analyzes Competition

To an economist, an assessment of the competitive impacts of a remedy proposal requires an assessment of the factors impacting on competition. Competition can be measured or understood in a variety of ways. One paradigm that is often used by economists is the Structure-Conduct-Performance paradigm. The Structure of an industry concerns such matters as the number of firms in a market and the market shares of firms in a market. For example, if an

---

[1] It would be the height of unfairness if the United States, having failed to submit its own economic models or analysis, were to respond to this model merely with criticism when it has refused to disclose its own economic models or analysis, if any, of the RPFJ's competitive impact, or lack thereof.

industry has twenty business firms, and no firm has more than a twenty percent market share, the industry is probably competitive. If the industry has only two firms, and one of the firms has an eighty percent market share, the industry is probably not competitive.

The Conduct of an industry refers to the behavior of business firms within an industry. How do they conduct business? Are they actively colluding? Do they frequently share price information? Does one firm normally set prices, while the other firms simply set the same price in response? These are all behaviors which may indicate lack of competition. Some of these behaviors may also be a violation of the antitrust laws.

Finally, the Performance of an industry refers to how well the industry serves the interests of consumers (or society generally). For example, are prices high or low relative to the costs of production? Is the quality of goods and services high or low relative to the cost of producing quality, and relative to what consumers are willing to pay for quality? Is the variety of goods and services high or low relative to the value which variety and choice have for consumers, and relative to the extra costs (if any) associated with producing and selling that variety?

Economists typically measure the interests of consumers using a concept called "consumer surplus" ("CS"). Consumer surplus is a dollar measure of the value which consumers receive by being able to purchase goods at a low price rather than a high price, by being able to purchase goods they want, and by obtaining good quality from what they purchase. For example, if a consumer would have been willing to pay $200 for an operating system, but only paid $50, then that consumer receives a consumer surplus of $150. If a second consumer would have been willing to pay $75, but only pays $50, then the second consumer receives a consumer surplus of $25.

Economists also typically evaluate the performance of an industry using a related concept called "total surplus" ("TS"). Total surplus is simply the sum of "consumer surplus" and "producer surplus" ("PS"). Producer surplus is a dollar measure of the value which producers receive by being able to sell their land, labor, or capital at a higher price rather than a lower price. For example, if a worker would have been willing to sell his labor for $35,000 a year, but is paid $50,000 a year, that worker receives a producer surplus of $15,000 a year. If a capitalist is willing to lend or invest his money for a 10% return, but receives a 25% return, that capitalist receives a producer surplus of 15%.

When an industry is competitive, its performance in terms of "total surplus" will be at a maximum. Its performance for consumers will also be near a maximum. When an industry is competitive, the only way to improve consumer surplus is to lower prices still further, but this would cause producers to suffer losses. Hence, when an industry is competitive, consumer surplus is at a practical maximum, because there must either be government subsidies or unhappy producers, if consumer surplus is to be increased still further.

When an industry is not competitive, its performance in terms of total surplus is reduced. When an industry is not competitive, prices are higher and output is lower, than what would occur if the industry were competitive. Because prices are higher, consumer surplus is lower, but producer surplus is higher. However, the total surplus is reduced, because the producer surplus is increased by less than the amount by which consumer surplus falls, so the sum of the two surpluses is reduced. Hence, whether we measure industry performance by the metric of "consumer surplus" or by the metric of "total surplus", more competition is better than less competition.

C. How an Economic Analysis Impacts this Case

Industry performance can be poor, either because the industry structure is bad, because the industry conduct is bad, or because both structure and conduct are bad. A well-designed competition policy would attempt to remedy or prevent both bad structure and bad conduct.

However, the antitrust law as it is presently formulated is not a well-designed competition policy. The antitrust law attacks bad conduct, but does not attack bad structure per se. A monopoly is usually a bad industry structure, which frequently leads to bad competitive performance, but a monopoly as such is not illegal under the antitrust laws. A monopoly is only illegal if it is acquired or maintained through anti-competitive conduct. Hence, even though Microsoft is a monopoly, if Microsoft never does anything illegal, Microsoft is perfectly free to record its monopoly profits at the expense of consumers.

However, Microsoft did act unlawfully.

It is, of course, the primary aim of the antitrust laws to protect consumers and competition, not competitors as such. Naturally, competition requires competitors, and consumers are better off when competitors are protected from certain types of anti-competitive conduct. Nevertheless, the interests of consumers are paramount when fashioning a remedy. The interests of competitors are of secondary importance. A disinterested economic analysis will always keep this goal in mind when comparing remedies for the Court's consideration.

D. Preliminary Data for the Economic Model

In order to be useful, an economic model must have as close a relationship to reality as possible given the constraints inherent in any model. An economic model cannot mimic economic reality entirely, because economic reality is too complex to model in its entirety, many aspects of economic reality are not humanly known, and such an exacting economic model would be far too complex for either humans or computers to calculate in a reasonably timely fashion. Hence, all economic models (like all scientific models) are a simplification of reality.

The first consideration is the basic economic data and assumptions. The primary data of interest are costs, revenues, profits, and market shares for each of Microsoft's three monopolies. These three monopolies are the Windows operating system monopoly, the Internet Explorer browser monopoly, and the Office (e.g., word processing, spreadsheet, and database) software monopoly. Each of these three monopolies is implicated in antitrust violations committed by Microsoft. The Windows operating system monopoly is especially implicated in these violations.

There is the question of whether we should model all three monopolies, or only one monopoly, for purposes of corrective action. This problem is solved by running one version of the model for Platform revenues only and another version of the model for all types of product revenue.

Neither Microsoft nor the DOJ has provided data on costs, revenues, and profits for each of Microsoft's three monopolies, or for any of them. The DOJ has not provided such data as part of its Competitive Impact Statement, nor has Microsoft provided such data on its Investor Relations website. However, Microsoft does provide data for revenues for various business units since July 1997.[2] These business units are "Desktop Platforms", "Desktop Applications", "Enterprise Software and Services", and a few other miscellaneous units. The "Platforms" unit corresponds most closely to Microsoft's operating system monopoly. The "Applications" unit corresponds most closely to its Office, and possibly its browser, monopoly. It is unclear at this time whether, and to what extent, the "Enterprise Software" unit corresponds to either competitive or monopoly markets, including operating systems for server markets, the browser market, or commercial services based on the Internet.

---

[2] Source: Microsoft's Investor Relations website, Microsoft Financial Statements, spreadsheets accessible from http://www.microsoft.com/msft/history.htm, downloaded December 5, 2001.

Hence, as initial data for the economic model, four sets of revenue figures for Microsoft's monopolies were used. The first set of revenue figures is based solely on Microsoft's Platform Revenues, which most closely conforms to a narrow vision of Microsoft's monopoly. The second set of revenue figures is a summation of Platforms & Enterprise Software. The third set of revenue figures is a summation of Platforms & Applications. The fourth set of revenue figures is a summation of Platforms, Applications & Enterprise Software. The revenue figures are arranged in increasing order of size, with the first set of figures being the smallest, and the fourth set of figures being the largest. This information is shown in Attachment A-2 which immediately follows this Attachment A-1.

As it turns out upon analyzing the results produced by the model, the qualitative conclusions of the economic model are basically unaffected by whether the model uses Platform revenues as a base or essentially all product revenues as a base. Quantitative results will change, of course, because the fourth set of figures roughly triples the calculated values compared with the first set of figures. Nevertheless, the qualitative conclusions remain the same.

In order to place these historical figures into useful format, the revenue figures are projected backwards in time through calendar year 1995. This is done by computing quarterly revenues for each business unit as a percentage of total revenues. A statistical regression on these percentages was used to determine if these percentages were growing or shrinking. These statistical tests indicated modest, but statistically significant, changes in these percentages over the time interval July 1997 through September 2001. Hence, similar percentage changes were used to determine the missing historical data for January 1995 through June 1997. These projected percentage changes for the three business units were multiplied by Microsoft's reported total quarterly revenues for the quarters of these prior years to obtain estimated values for the revenues of each of Microsoft's three main business units for each such quarter.

These data were converted from nominal dollars to real dollars. Nominal dollars are simply the actual reported dollars, without any adjustment for changes in purchasing power due to inflation. Real dollars are nominal dollars as of a given year, but adjusted for inflation for years other than the base year in which the real dollars are being reported. In order to convert the nominal dollars into real dollars, the U.S. Bureau of Labor Statistics' (BLS) Consumer Price Index (CPI) for "All Urban

Consumers (Current Series)" was used.[3] This is the most commonly used inflation index. The nominal dollars were converted to real dollars using 2001 as the base year.[4]

Next, the real quarterly revenues were projected into the future. For each of the three business units, the 1995-2001 historical growth rates were calculated using log-linear statistical regressions. Revenue growth rates were very high, 19.8% annual growth for Platforms, 18.5% annual growth for Applications, and 28.9% for Enterprise Software, all expressed in real dollars. However, revenues did falter a bit in the last year of data. Hence, I used the average of the last four quarters of the data available to me as the baseline to estimate the last quarter of revenue data for calendar year 2001.[5] Upon this baseline estimate of revenue for the fourth quarter of 2001, I projected all future growth.

In order to project future growth, I assumed that software

---

3 Downloaded December 5, 2001 from the BLS's CPI web page, available at http://stats.bls.gov/cpi/home.htm .

4 The average CPI for 2001 was computed as the eight-month, mid-year average for 2001. Since the last two months of 2001 were not yet available, the first two months were dropped for symmetry.

5 Microsoft's accountants use a fiscal year which differs from the calendar year. I re-dated all Microsoft figures to their true calendar years.

production would eventually become a "mature" industry. As a mature industry, real growth rates are unlikely to exceed some modest figure, such as 3% per year. However, computer software has not yet reached this stage of maturity. Software growth is very much driven by the phenomenal growth in computer hardware capabilities. The growth rate of computer hardware capacity is unlikely to taper off anytime soon, even if we restrict our attention to foreseeable technological developments.

However, revenue growth rates for software are unlikely to be sustained indefinitely into the future at annual rates of 18%-30%, no matter how amazing these future developments in computer hardware may be. Accordingly, I project that the current rapid growth in monopoly revenues will gradually slow down to the more modest growth rate of 3% a year. In my projections, I allow the historically-observed, rapid growth rates to converge towards the slower "mature industry" growth at the convergence rate of 5% per quarter. That is, if the growth rate in quarter 1 is 20%, then the growth rate in quarter 2 is assumed to be (20% x 0.95) + (3% x .05) = 19.15%. Alternative projections for Microsoft's future monopoly revenues may also be reasonable. However, it is unlikely that alternative projections will fundamentally alter the qualitative conclusions.

These quarterly estimates and projections for Microsoft's revenues by business division were then summed into annual figures for each calendar year from 1995-2025. Attachment A-2 provides the real revenue figures and projections which were used in the computerized economic model.

The next main piece of data is data on costs. Data on costs were also obtained from Microsoft's Investor Relations website.[6] Data on Microsoft's expenses are available for the company as a whole, but do not appear to be available by business division. Hence, the only option is to take an average across business divisions as being representative of Microsoft's three main business divisions.

Microsoft's spreadsheets available on the Microsoft website list their expense items as a percentage of revenue for each Microsoft Fiscal Year. The percentages from the last ten fiscal years were used to compute ten-year averages for each expense item as a percent of revenue. These 10-year averages are listed in Attachment B.

These expense items were then classified as either short-run costs or long-run costs. Microsoft's profit and loss sheet does

[6] Source: Microsoft's Investor Relations website, Microsoft Financial Statements, spreadsheets accessible from http://www.microsoft.com/msft/history.htm, downloaded December 5, 2001.

not show capital expenses as such. However, it does show Research and Development (R&D) expense. It is assumed that R&D for its software products is Microsoft's main long-term cost. "General and administrative" expense is also classified as a long-term cost. The other expenses I classify as short-run costs. According to this classification, Microsoft spends 41.01% of its revenue on short-term costs, and 18.55% of its revenue on long-term costs. These percentages have held fairly steady over the years, with some variations.

To the extent that long-term costs take time to develop their respective revenues, and to the extent that Microsoft's revenues are growing, these long-term costs as a percent of revenue are probably overstated. For example, if Microsoft's revenue in Year 1 is $100, and its R&D expense in Year 1 is $20, that is 20% of revenue. However, suppose that it takes 4 years for Microsoft's R&D expenditure to pay off. Suppose that in the same 4 years Microsoft's revenue has doubled to $200. Microsoft's $20 R&D expenditure in Year 1 has helped to create $200 of revenue in Year 5. This is a percent of revenue of only 10%, not 20%.

However, to the extent that investors require a positive return on their capital investments, these long-term costs as a percent of revenue may be understated. For example, if investors require a return of 50% on their capital over a 4-year period, then an investment of $20 in Year 1 will require repayment of $30 in year 5. If Microsoft's revenues had remained at $100 in Year 5, this would be a percent of revenue of 30%, not 20%. If Microsoft's revenues rose to $200 in Year 5, this would be a percent of revenue of 15%, and not 10%, 20%, or 30%.

For purposes of the computerized economic model, it is assumed that these two effects offset each other, and accordingly the model uses the raw percentage, 18.55%, as Microsoft's long-term cost of production. These two effects will exactly offset each other only if investors' required return on capital exactly matches Microsoft's growth rate. This is unlikely to happen exactly. It is most likely that the investors' required real rate of return on capital investment is less than Microsoft's phenomenally rapid growth rates in revenue. Hence, Microsoft's long-term cost of production is probably somewhat less than 18.55%.

Finally, we consider Microsoft's market share. In the Findings of Fact, Judge Jackson indicated that Microsoft's market share in operating systems was over 90% for over a decade.[7] More recent market share data indicates that Microsoft has approached or

[7] Finding of Fact number 35. U.S. v. Microsoft, 84 F.Supp.2d 9, 19 (D.D.C. 1999)

exceeded a 90% market share in all three of Microsoft's monopolies: Since the beginning of the trial, Microsoft's share of the web browser market has increased from less than 45% to more than 87%, its position in the desktop operating system market has risen to 92% (a 3% increase in the last year) and its market share for business productivity applications, such as word processing and e-mail, is now over 96%.[8]

E. Equations for the Economic Model

An economic model must model both the demand side and the supply side of the markets in question. However, to keep the model simple and tractable, it is best to use equations that are fairly easily solved and calculated. For the demand side, I assume that the product being produced is "homogenous". This means that the product is essentially the same, in the eyes of the consumer, whether the product is produced by one firm or another firm.

Software products produced by different firms are probably not completely homogenous, either because a firm's reputation, or its product quality, or other product features may differ across firms. However, the assumption of product similarity across firms is often

---

8 Edward J. Black, "*The Microsoft Monopoly and its Effects,*" Computer Und Recht International, April 2001.

true enough for modeling purposes. In addition, even though product quality may differ, a simple reinterpretation of the model can handle such situations. To the extent that people are willing to pay more for higher quality, we can interpret this situation as if the higher quality is equivalent to higher quantity.

Another simplifying assumption for the demand side is that the industry demand curve (graph of the price of a product vs. quantity of a product demanded at each price) is linear. A demand curve is unlikely to be linear (that is, it is unlikely to be a straight line). However, the only range of prices worth considering for the competitive analysis is the prices and outputs that lie between the monopoly price and output and the competitive price and output. Over a small range of prices and outputs, the demand curve is likely to be close to a straight line. Therefore, it is unlikely that assuming curvature or lack of curvature in the demand curve will play any significant qualitative role in the conclusions of such a competitive analysis.

Accordingly, the demand side assumes that products are homogenous and that demand curves are linear, according to the equation:

$$P = A - bQ \qquad (1)$$

Where P=Price (same for all firms), Q=Industry Output Quantity, and

A and b are positive parameters (intercept and slope of the demand curve).

We now turn to the supply side. Technically, only competitive firms have supply curves (graph of the price of a product vs. quantity of a product supplied at each price). Monopoly firms have only marginal cost curves. In this industry, we assume that firms are few in number, either one or a very few firms. Hence, the industry at all times is either a monopoly or an oligopoly.[9] Standard textbook theory tells us how to analyze the production decisions of a monopoly firm. However, there is no single textbook model for how to analyze an oligopoly. This is because there are multiple ways in which an oligopoly industry might behave.

In order to analyze the production decisions of either a monopoly or an oligopoly, it is necessary to posit the nature of the cost curves which they face. It is assumed that different firms may have different costs of production. However, for simplification, it is assumed that each firm (subscripted i for each firm i, where i = 1, 2, 3, ...) has both a fixed cost (Fi) and a marginal cost (Ci). It is assumed the marginal cost is constant (but different) for each firm. Since the fixed cost has an effect

9 An "oligopoly" is an industry structure where there are only a few firms (at least two), or only a few main firms.

only on entry decisions, exit decisions, and shut-down decisions, rather than pricing decisions, it is assumed that the fixed cost is the same for all firms ($Fi=F$ for all i). These simplifying assumptions are unlikely to have a significant qualitative impact on the conclusions.

Hence, the total cost or cost curve for each firm is assumed to be:

$$TCi = Fi + QiCi \qquad (2)$$

Where TCi = Total Cost for firm i, Fi = Fixed Cost for firm i, Qi = Quantity of output for firm i, and Ci is the constant marginal cost for firm i. In addition, we assume that $Fi = F$ for all firms which are producing and $Fi = 0$ for all firms which are not producing.

For a monopoly firm, it is sufficient to know the cost side and the demand side to obtain a prediction for the production decision. The monopolist's profit is:

$$\begin{aligned} Profit_i &= TRi - TCi \\ &= PQi - (Fi + QiCi) \\ &= PQi - QiCi - Fi \qquad (3) \end{aligned}$$

Where TRi = Total Revenue for firm i = PQi, and TCi comes from equation (2).

Assuming that the fixed cost is not so high as to make production not profitable, the monopolist finds it most profitable to produce at the output level where marginal cost (MC) equals marginal revenue (MR). On a graph showing a plot (or curve) of dollars of profit per unit vs. the quantity of units produced, this output level (where MC = MR) is the highest point on the curve. The eye can determine this point at a glance.[10] To determine this output level by computer, calculus is used and this output level is determined by obtaining the partial derivative of Profits with respect to the firm's choice of Qi and setting these derivatives equal to zero:

$$(d\ Profiti\ /\ d\ Qi) =$$

$$(d\ P\ /\ d\ Qi)Qi + P - Ci = 0$$

$$bQi + P - Ci = 0 \qquad (4)$$

Where we substitute (d P / d Qi) = b from the derivative of the demand curve in equation (1).

For an oligopoly firm, we must make a choice from many possible oligopoly models, a model which is reasonable for the situation at hand. A standard oligopoly model, first developed by a French economist named Cournot over 150 years ago, is still frequently used by economists today because it is fairly easy to

---

[10] In artistic terms, a picture is worth a thousand words. In mathematical terms, the eye is a better integrator than the mind.

compute. The Cournot model assumes that each oligopoly firm makes its output decision under the assumption that rival firms will not change their output in response to its own change in output. The Cournot model yields an oligopoly price and output which is intermediate between competition and monopoly. Also in the Cournot model (when firms have the same marginal cost), an increase in the number of firms causes prices to fall and output to rise. When there are a very large number of firms, the Cournot model predicts competitive pricing, which is what we would expect.

When all firms attempt to maximize their absolute level of profits, the profit-maximizing equations for each firm under the Cournot model are:

$$Profit_i = PQi - QiCi - Fi \qquad (4)$$

$$(d\ Profiti\ /\ d\ Qi) =$$

$$(d\ P\ /\ d\ Qi)Qi + P - Ci = 0$$

$$bQi + P - Ci = 0 \qquad (5)$$

The Cournot model is reasonable for the circumstances of this industry. Given a fairly significant level of fixed costs for this industry, it is unlikely that more than two or three firms can survive as major players in this industry. Fixed costs for software production (i.e., for research and development) require that firms must have significant sales simply to break even. This limits the number of firms which can survive as major players in the industry.

Microsoft's long-run costs appear to be about 18.55% of revenues. If all of these costs are fixed costs, then no more than five firms can exist in the industry, because fixed costs for six firms would eat up 18.55% x 6 = 111.3% of the industry's total revenue. This is unviable. In addition, there are also the short-run costs that must be covered. Furthermore, when there are two or more firms in the industry, we expect prices to fall, which allows firms to sell more, but only at a lower profit margin.

Computer results from a preliminary economic model, which allowed up to five firms in the industry, indicated that if fixed costs are either 75% or 100% of the long-run costs, then only two firms can survive in this industry. If fixed costs are either 25% or 50% of the long-run costs, then only three firms can survive in this industry. If fixed costs are 0% of the long-run costs (i.e., all long-run costs are variable costs), then it is possible for four or five firms to survive in this industry. Accordingly, the computer model was revised to consider a maximum of three firms in the industry.

Given that only two or three firms can successfully survive, under Cournot assumptions, we may ask if the Cournot model is a reasonable description. Alternative oligopoly models do exist, and these may suggest either higher prices or lower prices than what the Cournot model would predict.

Under the circumstances of an industry structure with only two or three firms, it is more reasonable to assume that prices may be higher than the Cournot model would predict. This is so for two reasons. First, software products are likely to be somewhat differentiated, rather than homogenous, as the computer model assumes. If products are differentiated, then consumers see the products of different firms as being somewhat different from each other, albeit also similar to each other. For example, Corel WordPerfect and Microsoft Word have their differences, as well as their similarities. Within a small range of prices, each software product can act as a kind of "mini-monopolist" with respect to its own product price.[11]

Second, when there are only two or three firms, tacit collusion which raises prices is easier to implement, and difficult to prove. Moreover, unlawful conspiracies to raise prices are less easily discovered. However, it is the general experience that oligopolies with very few firms rarely collude by means of unlawful

---

11 Economists normally use the term "monopolistic competition" rather than "mini-monopoly" to describe this type of phenomenon.

conspiracies (which could net jail time), presumably because tacit collusion is so much easier.

For both these reasons, it is substantially more likely that oligopoly prices would be higher than what the simple Cournot model would predict, than that the oligopoly prices would be lower. If we assume that prices would be higher, this means that more firms can survive in the industry. For example, if the Cournot model would predict that only one firm can be profitable, it may be that two firms can be profitable. If the Cournot model predicts only two firms can survive, it may be that three firms can survive. And so forth.

Hence, the Cournot model is probably a bit cautious in its predictions about how many firms can actually compete and survive in this industry. This is probably a good thing. One of the issues in this case, at least implicitly, is whether or not Microsoft is a "natural monopoly." If Microsoft is a natural monopoly, someone might argue, then Microsoft caused little or no harm by keeping out the competition, since the competitors could not have survived anyway. The computerized model does not in any way lend support to this type of argument. Hence, the Court should not be reluctant to consider structural remedies which divide Microsoft into two or more firms.

F. Equations For a Relative Profit Maximizing Firm

One of the options for a structural remedy is to change the incentives of the business managers of the successor firms to Microsoft when Microsoft is re-structured. The incentives of the business managers can be altered by changing the method of compensation for the officers of the business firm. A method of incentives for preventing collusion is further explained in a paper published in a refereed academic journal.[12]

For purposes of this comment and the computer economic model, attention is restricted to the simplest possible methods for implementing this incentive system. More complex methods for implementing the incentive system are certainly possible, and some of these more complex implementations may even be better or more effective than the simple implementation discussed here.

In its purest implementation, the incentive scheme sets up a zero-sum game for two or more firms in an industry.[13] In the zero-sum game, there is no incentive for all firms in the industry to engage in any type of collusion. The method even prevents tacit

12 Carl Lundgren, Review of Industrial Organization, Volume 11, Number 4, August 1996, pp. 533-550. A copy of this article is attached as Attachment T.

13 Simply put, RPM forces the firms in the industry to be cutthroat competitors which serves to open markets to competition which is the principal goal of the antitrust laws.

collusion, which may be hard to detect, and difficult or impossible to prosecute. The method accomplishes this amazing feat simply by changing the financial incentives of business managers, not by passing strict new antitrust laws with draconian penalties.

The method sets up a set of incentives called Relative Profit Maximizing (RPM) incentives. Business firms whose managers are motivated by these incentives may be called Relative Profit Maximizing (RPM) firms. Each business manager is assumed to be motivated by at least some desire to increase his wealth. In a well-run business firm, managers are normally paid in a manner which motivates them to increase their wealth by increasing the profits of their firm. RPM incentives alter these common methods of financial compensation by additionally motivating the manager to maximize the firm's profits relative to competing firms' profits.

In its most general form, the goal of the RPM manager is to maximize his profits relative to the profits of rival firm(s). It is only be achieving this goal that the RPM manager can attain maximum financial satisfaction, because that is how the manager is being paid. In its simplest form, the goal functions for two rival RPM firms look as follows:

$$\text{Goal1} = \text{Profit1} - z(\text{Profit2}) \qquad (6)$$

$$\text{Goal2} = \text{Profit2} - z(\text{Profit1}) \qquad (7)$$

When z = 1.0 in the above two goal functions, we set up the pure zero-sum game. In the zero-sum game there is no incentive to collude. If instead, z = 0.0 in the above two goal functions, then both firms are motivated by Absolute Profit Maximizing (APM) incentives. APM incentives are simply the incentives we normally expect to find in business firms. Absolute Profit Maximizing (APM) firms simply try to maximize their own level of profits, regardless of the level of other firms' profits. APM firms—which are the most common type of business firm in a capitalist economy—do have an incentive to collude, if an opportunity arises.

In simple terms in a two firm industry using RPM incentives, if a manager increases his firm's annual profits by 10% which is equivalent to $1 billion he only gets a bonus (or salary in the case of absolute dependence on RPM) if the profits of the other firm in the industry increase by less than 10%. In a two firm industry using APM incentives, the manager would get a bonus for the extra $1 billion even if his firm's profits increased less than the other firm's profits in terms of annual percentage gain.

The parameter z in the above goal functions can also take on additional values. For example, if z is set less than zero, the two firms would have Joint Profit Maximizing (JPM) incentives. JPM incentives would likely create less vigorous competition between the two business firms than would otherwise occur with APM incentives.

If z in the above goal functions is between 0.0 and 1.0, this creates an impure system of relative profit maximizing incentives. For example, if z = 0.3, this creates a mixture of two incentive schemes which might be described as "30% RPM plus 70% APM." An impure RPM incentive scheme partially reduces the incentive for collusion, but does not completely eliminate the incentive for collusion. An RPM firm, even one with an impure RPM incentive, can normally be expected to compete more vigorously than an APM firm. For this reason, the Court should consider using RPM incentives as part of an overall structural remedy.

For purposes of illustration with the computerized economic model, only values of z between -0.3 and 0.9 are used. Generally, z is in the range of 0.0 to 0.9 in the model and no preferred solution has z less than 0.0. The value of 1.0 (pure RPM) is avoided, because with this simple illustrative model (with no mechanism for avoiding losses), pure RPM would practically guarantee that one or both firms will lose money. This is because if the industry has little or no product differentiation, pure RPM causes prices to be set to the average of marginal costs. If in addition, software firms have high fixed costs, pure RPM practically guarantees that at least one firm, and possibly both firms, will be unable to recover their fixed costs of production. Pure RPM may still be useful and beneficial, but only if additional mechanisms are instituted to avoid this outcome.

The goal-maximizing outputs for the goal functions listed in equations (6) and (7) are:

$$(d\ Goal1\ /\ d\ Q1) =$$

$$(d\ P\ /\ d\ Q1)Q1 + P - C1 + (d\ P\ /\ d\ Q1)Q2 = 0$$

$$bQ1 + P - C1 + bQ2 = 0 \qquad (8)$$

$$(d\ Goal2\ /\ d\ Q2) =$$

$$(d\ P\ /\ d\ Q2)Q2 + P - C2 + (d\ P\ /\ d\ Q2)Q1 = 0$$

$$bQ2 + P - C2 + bQ1 = 0 \qquad (9)$$

G. Basics of Scenario Analysis

The purpose of a scenario analysis is to provide a projection of a range of possible futures. The basic parameters of an economic model are usually not known, although they can often be estimated (through empirical or theoretical analysis). These estimates may be arrived at with a greater or lesser degrees of confidence, accuracy, and reliability. Additionally, even if the basic parameters of an economic model were known with certainty, most economic models allow for uncertainty in how those basic parameters will vary for particular firms or individuals. For example, even if it were known with certainty that the probability of bankruptcy for a particular firm in a particular industry was exactly 3% a year, this would not tell us whether that particular firm will be bankrupt in twenty years.

In a well-done scenario analysis, one should vary the parameters through a reasonable range of values, including both moderate values and extreme values. In addition, the fate of individual firms (given the assumed parameters for a particular scenario) is varied according to the laws of probability governing that particular scenario.

There are two basic ways of conducting a scenario analysis. One way is to compute all the possibilities (appropriately weighted by probabilities) for a limited number of parameters that are allowed to vary through a small number of reasonable values for each parameter, including both moderate and extreme values. The second method is called a "Monte Carlo" study. The Monte Carlo study allows a large number of parameters to be varied, randomly, through a large set of possible values. The Monte Carlo study necessarily uses random numbers, which are available in many computer packages. The first type of study might or might not use random numbers.

The computer model used for these comments employs the first method of scenario analysis. Probabilities for every scenario are exhaustively computed and assigned. No random numbers or random number generators were used in the analysis.

The computer model computes probabilities and outcomes for two distinct types of scenarios. One type is a static scenario. The static scenario occurs at a particular period of time, within a single transition period. These transition periods (for a change or transition from one short run cost level to another as is discussed further in section H below) are assumed to have a length of three, five, or eight years.

The other type is a dynamic scenario, which is a path that links two or more static scenarios occurring in two or more time periods. For each set of initial conditions and basic parameters, the computer starts with a single scenario in transition period zero. The computer then calculates the probability that various additional static scenarios will be reached in transition periods one through ten. The probability that one static scenario will turn into another static scenario depends on how similar or dissimilar are the two scenarios. The computer calculates the outcomes for every static scenario, and weights those outcomes by the probability that the static scenario will occur in each of the eleven transition periods (periods zero through ten).

H. Details of Static Scenarios

The static scenarios assume that firms differ only by level of cost. The computerized economic model assumes that there are three firms and five levels of short-run cost. These five levels of cost are level one (lowest cost), level two, level three, level four, and level five (highest cost). These five levels of cost are assumed, over the long run, to have differing probabilities of occurrence. In particular, the probability of cost level one (lowest cost, 10% chance) is assumed be lower than the probability of cost level five (highest cost, 30% chance). This reflects the plausible assumption that it is easier to be a high-cost firm than a low-cost firm.

All possible combinations of the five cost levels for three firms are computed. These possible combinations are organized into 35 static scenarios. Whenever a static scenario has the same cost level for two or more firms, the costs of each firm are adjusted slightly so that no two firms have the same level of cost. A list of the cost levels associated with each static scenario is shown in Attachment C. The weighted average of cost levels over all firms and scenarios is 3.5.

The basic parameters for static scenarios are varied along two dimensions. The first dimension is the cost spread for short-run costs. The cost spread is defined as the ratio of cost level one to cost level five. For example, if the lowest cost level is twice as efficient as the highest cost level, then the cost spread is 50%. Five different ratios for the cost spread were chosen for the analyses. These cost spread ratios were 25%, 33%, 40%, 50%, and 67%.

The second dimension for variation is the portion of long-run cost which is allocated to fixed cost. The portion of long-run cost which is actually a fixed cost is open to some question or interpretation. The mere fact that a software firm has spent $X billion on software development does not mean that the whole expenditure was necessary to develop the software in question. Five different values for the fixed-cost portion of long-run costs were computed. These percentages were 0%, 25%, 50%, 75%, and 100%. In all cases, the remainder of the long-run cost was classified as a variable cost.

Thus, twenty-five static variations on the basic parameters were computed. For each of these variations, the computer programs computed the prices, quantities, profits, and consumer surplus outcomes for each of the thirty-five static scenarios. These static numbers were applied to the probabilities computed for each static scenario for each of the eleven transition periods. The computer model uses the static figures and the associated probabilities for each transition period to compute the expected profit and consumer surplus outcomes for each transition period.

I. Details of Dynamic Transitions

The basic parameters for determining the probabilities of transition effectively vary along only one dimension: The speed with which transitions occur from one cost level to another. This speed variable is implemented in two different ways.

The first method is relatively straightforward. The length of time for the transition periods is allowed to vary. A three-year length for the transition period implies a fast transition speed. A five-year length implies moderate transition speed, and an eight-year length implies a slow transition speed.

The second method influences the speed of transition by determining the extent by which one static scenario may change into another static scenario, from one transition period to the next transition period. For all transition speeds, the model assumes that one static scenario is more likely to change to another static scenario, the more similar are the two scenarios. The measure of similarity or dissimilarity between two scenarios is determined by how similar or dissimilar the short-run costs are for each firm in the industry.

In the slow speed for transition, the second method presumes that a firm's short-run cost cannot change more than one level at a time. For example, a firm whose cost level is four, can change to cost levels five or three, and it can stay at cost level four, but it cannot move to cost levels one or two in only one period of transition. In the slow transition, the firm is more likely to stay at the same cost level, from one transition period to the next, than to move to the cost level above or below.

In the moderate speed for transition, the second method presumes that a firm's short-run cost cannot change more than two levels at a time. For example, a firm whose cost level is four, can change to cost levels two, three, or five, and it can stay at cost level four, but it cannot move to cost level one in only one period of transition. In the moderate speed transition, the firm is more likely to move only one cost level, rather than two cost levels, from one transition period to the next.

In the fast speed for transition, the second method presumes that a firm's short-run cost can change as many as four levels at a time. For example, a firm whose cost level is one, can change to cost levels two, three, four, or five, and it can also stay at cost level one. In the fast transition, a firm is more likely to move only one cost level than two cost levels, more likely to move two levels than three levels, and more likely to move three levels than four levels, from one transition period to the next.

The computer model also causes the exit of firms from the industry when their short-run costs become too high. If a firm's short-run costs reach the adjusted cost level of five or greater, the firm is presumed to exit the industry. This is because an experienced firm which cannot keep its costs down (or quality up) has no competitive advantage over potential competitors, and has presumably lost its ability to compete profitably. The model presumes that the exiting firm is replaced by a new firm which is equally high cost. The new firm then has the opportunity to reduce its cost in future transition periods. Hence, all new entrants to the industry are presumed to enter with high short-run costs.

The computer model starts transition period zero, either with Microsoft as a monopoly, or with Microsoft divided into two or three firms. If Microsoft starts as a monopoly, Microsoft is presumed to start at cost level three. Cost level three is midway between cost level one (lowest cost) and cost level five (highest cost). Cost level three is slightly more efficient than the long-term average cost level of three and a half. Although some may argue that Microsoft acquired its monopoly because it was so much more efficient than its competitors, that monopoly acquisition happened at least ten years ago and was probably due to the arguably per processor licensing which was the subject of a prior consent judgment (attached as Exhibit 2 to this comment letter). There is no reason to suppose, today, that Microsoft has anything other than about average efficiency for an incumbent firm.

If Microsoft is split into two or three firms, we may suppose that there could be some cost-efficiency losses due to initial disorganization. To see this possibility in extremis, suppose that the Court ordered Microsoft divided into ten competing firms. We might consider ourselves lucky if three of the ten firms were equally efficient as Microsoft is today. However, we should not exaggerate the likely cost-inefficiency impacts of dividing a very large company into two or three very large companies. If Microsoft is split into two firms, the model assumes that one of the Microsoft successor firms starts at cost level three, while the other starts at cost level four. If Microsoft is split into three firms, the model assumes that one of the Microsoft successor firms starts at cost level three, while the other two successor firms start at cost level four.

The computerized economic model also treats the initial period (period zero) differently from the subsequent transition periods. In the initial period, potential competitors do not produce; only Microsoft or Microsoft's successors produce. In subsequent periods, both Microsoft and competitors can produce. This is because, at least initially, major competitors do not exist, because their entry has been blocked by anti-competitive acts. However, under the presumption that an effective conduct or structural remedy creates the opportunity for entry, competitors can produce in subsequent transition periods.

J. Construction and Computation of Remedy Alternatives

The computerized economic model developed for these comments is best suited for analyzing structural remedies. Nevertheless, the model can be applied to analyze conduct remedies, albeit with some caveats.

The model computes several alternative basic outcomes for the industry. The first basic alternative is "no remedy". If there is no remedy, it is assumed that Microsoft is a monopoly in all years from 1995 through 2025.

The second basic alternative is a 100% effective conduct remedy, starting in 2002. To calculate the results in terms of CS, TS, and the profits of Microsoft and its competitors in the case of a 100% effective conduct remedy, the model assumes all barriers to entry are removed and there is no anti-competitive conduct in the market. Under the assumption that there are no barriers to entry into the market, Microsoft starts as a monopoly in 2002, but is subject to entry from competitors thereafter. The choice of an early date for a conduct remedy is due to the timing of the negotiated conduct remedy, or alternatively the timing of the conduct remedy offered by the Litigating States. Hence, either conduct remedy can go into effect almost immediately.

In practice, no conduct remedy is likely to be 100% effective. The Litigating States' strong conduct remedy may be perhaps 60% to 80% effective as a conduct remedy.[14] The DOJ's weak conduct remedy may be about 20% effective. If we optimistically assume that the DOJ has hidden all the convincing and persuasive evidence which should have been in the Competitive Impact Statement, the DOJ might someday provide evidence to the public and the Court that the negotiated agreement with Microsoft may be 40% effective.

The model does not specifically compute the effects which any particular provision of a conduct remedy may have on future

---

[14] The "Litigating States" are the District of Columbia, California, Connecticut, Florida, Iowa, Kansas, Massachusetts, Minnesota, Utah, and West Virginia.

competition. Rather, it is up to the Court or the analyst to subjectively assess the overall effectiveness of a particular proposed conduct remedy, and to judge it accordingly. The computer model simply combines the two basic alternatives, "no remedy" and "100% effective conduct remedy," to compute estimated outcomes for conduct remedies with only partial effectiveness. For example, to compute a "60% effective conduct remedy" the program computes a weighted average of the two basic remedies, with a 60% weight on "100% effective conduct remedy" and a 40% weight on "no remedy." The outcomes in the case of other partially effective remedies are calculated in a similar manner.

The third set of basic alternatives is a structural remedy in which Microsoft is divided into two or three competing firms. If we accept the DOJ's pessimistic appraisal, no structural remedy can reasonably go into effect before 2005. More optimistically, if the Court follows the road maps laid out by the Appeals Court and the Supreme Court, there is at least a 50% chance that a structural remedy could take effect in 2003, without such remedy being overturned or stayed. In any case, the computer model pessimistically assumes that a structural remedy is not available before 2005. This time delay somewhat disadvantages the structural remedy, but the structural remedy is sufficiently superior to the conduct remedy, that it is not much of a disadvantage. Without the time delay, a structural remedy would always be superior to a conduct remedy.

The model computes several variations on a structural remedy. The first main variations are the division of Microsoft into two or three absolute profit maximizing (APM) firms. An APM firm is simply the conventional profit-maximizing firm that we see everyday in the business world. This type of division of Microsoft into two or more firms has been advocated by several economists, including four economists who filed an amicus brief before this Court.[15]

The second main variations are the use of relative profit maximizing (RPM) incentives after Microsoft is split up into two firms. A primary advantage of the RPM incentives is that competition can be maintained even if there are only two RPM firms in the industry. RPM incentives can be applied to two firms, three firms, or even more firms, but this computer model only applies RPM incentives to two Microsoft successor firms. The RPM incentives are assumed to be in effect so long as both Microsoft successor firms are still in the industry. If either RPM firm exits the

---

15 Robert E. Litan, Roger G. Noll, William D. Nordhaus, Frederic Scherer, "Remedies Brief of Amici Curiae," United States v. Microsoft Corp., filed with District Court, April 27, 2000.

industry, the goal function of the remaining Microsoft successor firm returns to the usual APM incentives.

The computer model also prints out estimates for the two-monopolies remedy previously proposed by the Plaintiffs in this case. If Plaintiffs' remedy worked as planned, it would be akin to a conduct remedy with enhanced effectiveness. In addition to removing the applications barrier to entry, the proposal would possibly introduce some measure of extra competition, because the two monopolists might decide to compete with each other. The computer model does not specifically analyze this remedy, but simply estimates its value as being a third of the distance between a "100% effective conduct remedy" and a 2-firm APM structural remedy. This is calculated as a weighted average of these two basic remedies, with a 2/3 weight on the "100% effective conduct remedy" and a 1/3 weight on the 2-firm APM structural remedy.

Finally, the model computes what might have happened along a "Lawful Path." The lawful path assumes that Microsoft starts as a lawful monopoly in the year 1995, and commits no antitrust violations at any time. Although some private lawsuits allege antitrust violations which occurred before 1995, this case does not concern those allegations. This case concerns anti-competitive acts committed by Microsoft in the browser wars, which did not start until 1995.[16] To simulate the lawful path, Microsoft starts as a monopoly in 1995, but is subject to potential competition in 1996 and later years.

The purpose of calculating the "Lawful Path" is to serve as an equity standard for evaluating alternative remedies. The Lawful Path tells us what Microsoft likely would have earned, if Microsoft committed no violations. To the extent that Microsoft's profits exceed those lawful earnings, we may refer to those excess earnings as the fruits of its unlawful actions. Likewise, to the extent that consumer surplus exceeds (or falls short) of what would occur along the Lawful Path, this is the extent to which consumers benefit (or remain harmed) as a result of a particular remedy.

K. Weighting of Alternative Scenario Parameters

The computerized economic model computes and weights 225 sets of scenarios, which differ by the basic parameters assumed for each scenario. These differ along four dimensions. Not all scenarios are equally likely. Hence, in the reporting of results, they are weighted by their likelihood of occurring. Attachment D shows the four basic parameters, the sixteen parameter values, the point

---

16 See Findings of Fact, Number 17. U.S. v. Microsoft, 84 F.Supp.2d 9, 14 (D.D.C. 1999)

values of their weighting, and their implied probability of occurring.

The first dimension of parameter variation is the cost-spread for short-run cost. Five different ratios for the cost spread were used: 25%, 33%, 40%, 50%, and 67%. In Attachment D these are labeled "Cost-Spread Ratio."

Studies of production efficiency between firms suggest that some firms can be only half as efficient as other firms. So that cost-spread ratios of 50% and 67% are certainly within the realm of plausibility. In addition, the short-run cost variable is doing double duty as a stand-in for possible differences in software quality between firms. If we assume similar ratios for differences in quality, then a 25% cost spread is certainly possible, though less likely. Such a cost spread implies that the inefficient firm has both double the costs and half the quality; it is an unlucky combination of extremes that is therefore less likely. Hence, I weight the 25% and 33% cost-spread ratios with a point value of 1, and weight the 40%, 50%, and 67% ratios with a point value of 2.

The second dimension of parameter variation is the portion of long-run cost which is allocated to fixed cost. Five different percentages for the fixed-cost portion were used: 0%, 25%, 50%, 75%, and 100%. In Attachment D these are labeled "Fixed-Cost

Portion of Long-Run Cost." At this point, there is no particular reason to suppose that one allocation of the fixed-cost portion is better than another. Hence, I assume a uniform distribution over the interval, 0% to 100%. This implies assigning point values of 1 to the two extremes (0% and 100%) and a point value of 2 to the in-between values (25%, 50%, and 75%).

The third and fourth dimensions for parameter variation are the speed of transition and the length of transition periods. In Attachment D "Transition Speed" takes on values of 1.5 (slow), 2.5 (moderate), and 4.5 (fast). In Attachment D, the length of transition periods is given by "Transition Length" of 3 years (short), 5 years (moderate), or 8 years (long). Extremes may either amplify each other (e.g., slow and long) or offset each other (e.g., slow, but short). Hence, even if we do not weight these values further, moderate combinations are more likely than genuinely extreme combinations. Hence, all transition speeds and transition length receive the same point value of 1.

Finally, for each of the 225 combinations of parameters, the points assigned to each parameter value are multiplied together. This yields a total of 576 points. Based on these point values, the computer model assigns each combination of parameter values an assumed probability of occurrence. These probabilities are used to weight the outcomes of the various calculations when reporting the final results, which we come to shortly.

L. Method of Computing Remedy Alternatives

For each remedy alternative, dollar values for costs, revenue, profits, and consumer surplus are calculated by the model in real dollars for each of the years, 1995-2025. These dollar values are calculated in real terms, in dollars of constant purchasing power, as of the year 2001.

It is generally standard practice to assume that money has at least some time value. That is, a dollar now is preferable to a dollar ten years from now, even if both dollars otherwise have the same purchasing power. One reason people prefer the dollar now is that money can be invested and earn interest. Another reason is that people are impatient.

In regulatory analysis of U.S. government regulations (e.g., under Executive Order 12886), it is standard practice to use a 7% real discount rate. This discount rate is somewhat akin to an interest rate. This means that future dollars will be discounted compared with present dollars, while past dollars will accumulate interest compared to present dollars.

Attachment E provides an example of how the 7% real discount rate can be applied to Microsoft's real monopoly revenues. In Attachment E it can be seen that Microsoft's revenues for its Windows monopoly were rather small, compared to what they will be if Microsoft operating systems continue to be a rapidly growing monopoly. In 1995 Microsoft's revenues for its Windows monopoly were only $3.0 billion in 2001 dollars. In 2002 they were estimated at $9.1 billion. In 2025 they are projected to be $34.1 billion in 2001 dollars.

When we apply the 7% discount factor, the picture changes somewhat. Revenues for 1995 "earn interest" of 50% when brought to 2001, while revenues from 2025 are discounted 80.3% from the value of equivalent purchasing power in of 2001. Discounted revenues for 1995 become larger ($4.5 billion) while discounted revenues for 2025 become smaller ($6.7 billion). The projected undiscounted revenues always grow, but the discounted revenues are projected to reach a peak in 2008, with $17.1 billion in undiscounted revenues and $10.65 billion in discounted revenues.

This illustrates an important cause of one of the more interesting results which emerge from the economic analysis: Because Microsoft's monopoly revenues are growing rapidly, we may anticipate worse damage to consumers in the future than what has already occurred in the past. Attachment F provides some comparisons for the Windows operating system monopoly which illustrate this result.

In Attachment F, the values for consumer surplus, competitors' profits, Microsoft's profits, and for the sum of these, total surplus, are provided for the past (1995-2001), the future (2002-2025), and in total (1995-2025). The top half of the Attachment shows the aggregated values of these quantities. The bottom half shows how these quantities compare with the same quantities along the Lawful Path. All quantities from the past earned interest at 7% per year, while all quantities from the future are discounted at the rate of 7% per year back to 2001 dollars.

Looking at the top half of Attachment F, we see both past and future values for "No Remedy," a "100% Effective Conduct Remedy," and a "3-firm APM Structural Remedy." In all cases, the future values for consumer surplus, Microsoft profits, and Total Surplus are substantially larger in the future, than in the past. In all these cases, the future values are more than double the size of the past values, even though the future is discounted and the past is inflated.

In the middle of Attachment F, we see the aggregated values for Lawful Path. Again all the future values are at least double the past values. If Microsoft had always pursued the Lawful Path, its profits would be lower, both in the past and in the future. Even on the Lawful Path, Microsoft's future profits are more than double its lawful past profits. Again, this is true even though past profits are inflated and future profits are discounted.

In the bottom half of Attachment F, the various aggregates in the top half of the Attachment are compared with the Lawful Path. If we compare "No Remedy" with the Lawful Path, we see very interesting differences between past and future. These differences are on the order of 10 to 1. Consumers in the past lost $4.1 billion in consumer surplus, but are scheduled to lose $35.0 billion in the future. Competitors lost $2.6 billion profit in the past, but are scheduled to lose $31.5 billion profit in the future.

Microsoft, by contrast, does extremely well. Microsoft gained $6.7 in unlawful extra profit in the past, but is scheduled to receive $60.4 billion in unlawful extra profit in the future. These numbers should give the Department of Justice and the Court some pause before adopting any settlement which effectively endorses continued extraction of profits from consumers due to anti-competitive conduct by Microsoft.

We may compare these numbers with what may happen under two alternative remedies. In the past (which no remedy can change), consumers lost $4.1 billion. In the future, they will lose an additional $4.7 billion under a 100% effective conduct remedy, but only an additional $0.3 billion under a 3-firm structural split-up of Microsoft Corporation.

Under a 100% effective conduct remedy, competitors in the future will still lose $6.7 billion while Microsoft gains $9.1 billion, relative to the Lawful Path. By contrast, under the 3-firm structural remedy, competitors lose $26.4 billion in the future, while Microsoft gains $26.5 billion in the future, relative to the lawful path. In other words, competitors benefit more from a 100% effective conduct remedy, while both consumers and Microsoft gain more from a structural remedy. This is an amazing result, which has some startling implications for how best to resolve this case.

This result does not appear to be an artifact of making peculiar assumptions in the economic model. The result is most likely due to the limited space available in the market for more than two or three firms. If Microsoft remains intact, competitors have room to enter the market and earn profits. However, if Microsoft is split into two or three firms, there is less room in the market for competitors to enter. Accordingly, under a structural remedy, the Microsoft successor firms all presumably initially owned by current Microsoft shareholders earn much of the profits which competitors might otherwise be able to take away.

This does not mean, as a practical matter, that competitors are necessarily better off with a conduct remedy than with a structural remedy. In actual practice, a pure conduct remedy cannot be 100% effective. A weak conduct remedy might be worse for competitors, while a strong conduct remedy may be better for competitors, as compared with a structural remedy. Likewise, Microsoft is not necessarily worse off with a conduct remedy than with a structural remedy. In comparison with a structural remedy, Microsoft may fare better with a weak conduct remedy than with a with a strong conduct remedy

M. Analysis of Computed Remedy Alternatives

Attachments G, H, I, and J provide a summary of the computations for several remedy alternatives. These summaries provide estimates of consumer surplus, profits for both Microsoft and its competitors, and total surplus. Total surplus is simply the sum of consumer surplus and the profits of all firms in the industry. These figures are aggregated for all the years, 1995-2025. They are expressed in real dollars, as of 2001. They are also appropriately discounted to the year 2001 at the standard 7% real discount rate which is commonly used in the analysis of United

States government regulations.

The first page of each of these Attachments provides the total values for each of the quantities, Consumer Surplus, Competitors' Profits, Microsoft's Profits, and Total Surplus. These figures are computed and summarized for each of the alternative circumstances. These circumstances are "No Remedy," conduct remedies with various levels of effectiveness, thirteen structural remedies which split Microsoft into two or more firms, and the "Lawful Path" in which Microsoft never disobeyed the antitrust laws.

The second page of each of these attachments compares each of the alternative circumstances with the Lawful Path. These numbers are calculated by subtracting the total quantities under the Lawful Path from the total quantities available under each alternative circumstance.

For example, in Attachment G Consumer Surplus under the Lawful Path is $105.9 billion, but under "No Remedy" the Consumer Surplus is only $66.7 billion. On the second page of the Attachment these two numbers are subtracted, so that we can see that consumers were/will be deprived of $39.2 billion in consumer value, if there is no remedy. Likewise, Microsoft has obtained/will obtain $108.5 billion under "No Remedy", but would have obtained only $41.3 billion under Lawful Path. The difference of $67.2 billion in profit is shown on the second page of the Attachment. This figure is representative of the unjust gain (the fruits of Microsoft's unlawful conduct) that Microsoft has obtained or will obtain if there is no remedy.

Attachments G and I calculate the remedy alternatives under the assumption that the only monopoly of concern is the Operating System ("Desktop Platforms") monopoly. Attachments H and J calculate the remedy alternatives under the assumption that all of Microsoft's monopolies ("Platforms" + "Applications" + "Enterprise Software") are of concern. The figures in Attachments H and J are approximately three times as large as the figures in Attachments G and I.

Clearly, "No Remedy" is not an option for this Court. These attachments also provide bottom line information on various conduct and structural remedies which the Court is entitled to consider. The first eight remedies are conventional remedies of a conduct or structural variety. In all four attachments, it may be seen that "APM, 3-firms" is the best of the conventional (non-RPM) remedies in the sense that best means maximum CS or TS. The "APM, 3-firms" remedy is simply a split-up of Microsoft Corporation into three competing successor companies, of the ordinary absolute profit maximizing (APM) variety. The three-firm split-up is similar to what other economists have advocated.[17]

We may confirm this conclusion by reading the first nine entries in the columns for "Consumer Surplus" and in the columns for "Total Surplus," on either the first or second page of each attachment. Of the first nine entries, the 3-firm APM remedy always has the largest consumer surplus, and also has the largest total surplus. It may also be noted that this 3-firm remedy restores most, but not all, of the consumer surplus and total surplus that would otherwise be wrongfully taken by Microsoft. This may be seen by the negative numbers for this remedy on the second page of each attachment.

Also of note for the 3-firm structural remedy is that Microsoft profits considerably from its unlawful acts, relative to the Lawful Path. This may be seen from the large positive numbers

17 See, for example, Robert E. Litan, Roger G. Noll, William D. Nordhaus, Frederic Scherer, "Remedies Brief of Amici Curiae," United States v. Microsoft Corp., filed with District Court, April 27, 2000; Lenard, Thomas M., "Creating Competition in the Market for Operating Systems: A Structural Remedy for Microsoft," Progress and Freedom Foundation Paper (January 2000); and Lenard, Thomas M., "Creating Competition in the Market for Operating Systems: Alternative Structural Remedies in the Microsoft Case," Progress and Freedom Foundation Paper (November 2000). These five economists advocate the three-firm split-up for Microsoft's Operating System company, and a separate company for Microsoft's applications.

for Microsoft's Profits for this remedy on the second page of each attachment. For Attachments G and I, Microsoft achieves an unlawful gain of $33.2 billion, even with the 3-firm split up. For Attachments H and J, Microsoft achieves an unlawful gain of $96.2 billion. Most of these remaining unlawful profits come from the pockets of competitors and would-be competitors (many of whom are not identifiable) who were excluded or deterred from competition by Microsoft's anti-competitive acts.

The consideration of structural remedies involving relative profit maximizing (RPM) incentives is as follows. In all cases, the RPM remedy is applied to only two Microsoft successor firms, after Microsoft is split into two competitors. These are shown in the attachments as "RPM, z=0.000" through "RPM, z=0.900". "z" is the value of the parameter z in the RPM firm's goal function. "z" tells us the extent to which a firm's business managers are financially motivated to maximize the relative profits of their business firm, rather than absolute profits. If z=0.0, there is no RPM incentive. If z=1.0, managers are solely motivated to maximize relative profits. For purposes of these comments, only the outcomes for values of z generally between 0.0 and 0.9 are illustrated. However, in the scenarios shown on Attachments I and J which allow a change in z (referred to as zbump=0.3) some percentage of all scenarios listed as having z values from 0.0 to 0.2 will have a z value of less than 0.0.

Attachments G and H assume that the value of z remains fixed, and that it does not respond to changing circumstances. In both attachments, consumer surplus is maximized when z=0.4 and total surplus is maximized when z=0.5. In Attachment G, the RPM solution can improve consumer surplus by $2.9 billion, and can improve total surplus by $4.6 billion over the 3-firm split up, which is the best conventional remedy. In Attachment H, the RPM solution can improve consumer surplus by $8.6 billion, and can improve total surplus by $13.5 billion over the 3-firm split up.

Attachments I and J assume that the value of z is more flexible, and can change in response to changing circumstances. The circumstance to which z is allowed to respond is the circumstance where one (or both) RPM firms are experiencing losses. These losses, of course, should not simply be short-term or even annual losses, but losses that are more chronic or long-term. In these computer runs, z is allowed to vary through a small range of values. In these attachments, z was allowed to range from the indicated value of z down to the smaller value of z which is 0.3 lower.

In Attachments I and J, consumer surplus is maximized when z=0.6, but this line includes some scenarios which can range down to z=0.3 due to the effect of a change in z as large as 0.3 (i.e., zbump = 0.3). Total surplus is maximized when z=0.8, but can range down to z=0.5 in the same manner due to a change in z as large as 0.3. In Attachment I, the RPM solution can improve consumer surplus by $9.3 billion, and can improve total surplus by $5.2 billion over the 3-firm split up, which is the best conventional remedy. In Attachment J, the RPM solution can improve consumer surplus by $27.2 billion, and can improve total surplus by $15.2 billion over the 3-firm split up.

In each of the Attachments, the 2-firm RPM remedy also reduces Microsoft's unlawfully acquired profits by a few billion dollars, relative to what Microsoft would obtain from the conventional 3-firm APM remedy. Hence, in all respects, whether measured in terms of increasing consumer surplus, increasing total surplus, or in the diminution of Microsoft's unjust fruits of its unlawful conduct, the RPM incentive system is capable of doing better than the best of the conventional economic remedies (APM).

N. Equity Analysis in Light of the Economic Analysis

The primary objectives of the antitrust laws, expressed in economic terms, is either to maximize consumer surplus or to maximize total surplus (or perhaps both, though it may not be possible to maximize both simultaneously). The Court should select a remedy according to whichever objective best fits the equity requirements of the antitrust law. According to the economic analysis just provided, a structural remedy combined with an RPM incentive, is better than any conventional structural or conduct remedy. Among the conventional remedies, the 3-firm split-up is better than any conceivable conduct remedy, including even a 100% effective conduct remedy. And, of course, among the conduct remedies, a strong conduct remedy (such as the Litigating States have proposed) is better than the weak conduct remedy which the DOJ has proposed.

A secondary objective is to assure that Microsoft does not gain extra profit in the future as a result of the future effect of its past (and continuing) unlawful behavior.

The computerized economic model (whose source code is attached as Attachments K-S) only models the price effects of Microsoft's anti-competitive acts. An additional harm caused by Microsoft in this case includes losses of innovation in the software industry.

Due to the failure of the United States to address this issue analytically in the CIS resource constraints precluded modeling these additional losses in consumer surplus and total surplus. It is possible that the dollar value of this damage to the consuming public (in the form of innovation which did not occur) caused by Microsoft's unlawful conduct exceeds the unlawful profits calculated by the model. Thus, it is unlikely that consumers and the public will ever regain that to which they are entitled as a matter of equity.

Attachment A-2

Microsoft Corporation
Annual Revenue by Business Division
Real 2001 dollars (in billions)

Calendar Year	Desktop Platforms	Platforms & Enterprise	Platforms & Applications	Platforms, Applications & Enterprise
1995	3.003532	4.207892	6.855180	8.059541
1996	3.727131	5.347350	8.460443	10.080662
1997	5.035883	7.458570	11.217205	13.639892
1998	6.454595	9.391382	14.204543	17.141330
1999	7.693463	12.210871	17.149630	21.667038
2000	8.186612	13.192271	17.729841	22.735500
2001	7.204304	11.348110	16.786433	20.930239
2002	9.142475	14.955194	19.792872	25.605591
2003	10.588836	17.711490	22.821122	29.943776
2004	12.001410	20.453823	25.771053	34.223466
2005	13.364562	23.136215	28.613009	38.384662
2006	14.670118	25.728773	31.332338	42.390993
2007	15.915669	28.215390	33.925832	46.225553
2008	17.102945	30.590792	36.398387	49.886235
2009	18.236419	32.857597	38.760186	53.381363
2010	19.322209	35.023736	41.024499	56.726025
2011	20.367255	37.100384	43.206070	59.939200
2012	21.378745	39.100399	45.319994	63.041648
2013	22.363742	41.037214	47.380985	66.054457
2014	23.328943	42.924094	49.402933	68.998085
2015	24.280557	44.773678	51.398672	71.891793
2016	25.224251	46.597717	53.379885	74.753351
2017	26.165138	48.406960	55.357103	77.598925
2018	27.107806	50.211128	57.339763	80.443086
2019	28.056353	52.018953	59.336296	83.298896
2020	29.014438	53.838247	61.354226	86.178035
2021	29.985331	55.675990	63.400283	89.090942
2022	30.971962	57.538426	65.480504	92.046967
2023	31.976974	59.431156	67.600333	95.054514
2024	33.002757	61.359227	69.764708	98.121178
2025	34.051497	63.327217	71.978146	101.253866

Source: Computed from spreadsheet data provided on Microsoft's Investor Relations website (downloaded December 5, 2001 from http://www.microsoft.com/msft/history.htm) and CPI indices from the BLS website (Downloaded December 5, 2001 from http://stats.bls.gov/cpi/home.htm).

Attachment B

Microsoft Corporation
Profit & Loss Items
As a Percent of Revenue
Ten-Year Average of Percentages
(Microsoft Fiscal Years 1992-2001)
& Classification of Expense Items
Into Short-Run and Long-Run Cost.

Profit & Loss Item	As a Percent of Revenue FY 1992-2001 ten-year average	
Revenue	100.00%	Categorized as:
Operating expenses:		
Cost of revenue	18.51%	Short-Run Cost
Research and development	14.73	Long-Run Cost
In-process R&D	0.19	Long-Run Cost
Sales and marketing	22.14	Short-Run Cost
General and administrative	3.63	Long-Run Cost
Other expenses	0.36	Short-Run Cost
Total operating expenses	59.59	
Total Short-Run Cost	41.01%	
Total Long-Run Cost	18.55%	

Operating income	40.41%
Losses on equity investees and other	-0.55
Investment income	6.04
Noncontinuing items	-0.27
Income before income taxes	45.97
Provision for income taxes	15.67
Net income	30.29

Source: Computed from spreadsheet data provided on Microsoft's Investor Relations website (downloaded December 5, 2001 from http://www.microsoft.com/msft/history.htm).

Attachment C

Adjusted and Unadjusted Cost Levels
For Firms in 35 Static Scenarios
And Long-Run Probability of Scenario

Static Scenario	Long-Run Probability	Unadjusted Cost Levels Firm1	Firm2	Firm3	Adjusted Cost Levels Firm 1	Firm 2	Firm 3
1	2.7000%	5	5	5	4.7500	5.0000	5.2500
2	6.7500%	4	5	5	4.0000	4.8333	5.1667
3	5.6250%	4	4	5	3.8333	4.1667	5.0000
4	1.5625%	4	4	4	3.7500	4.0000	4.2500
5	5.4000%	3	5	5	3.0000	4.8333	5.1667
6	9.0000%	3	4	5	3.0000	4.0000	5.0000
7	3.7500%	3	4	4	3.0000	3.8333	4.1667
8	3.6000%	3	3	5	2.8333	3.1667	5.0000
9	3.0000%	3	3	4	2.8333	3.1667	4.0000
10	0.8000%	3	3	3	2.7500	3.0000	3.2500
11	4.0500%	2	5	5	2.0000	4.8333	5.1667
12	6.7500%	2	4	5	2.0000	4.0000	5.0000
13	2.8125%	2	4	4	2.0000	3.8333	4.1667
14	5.4000%	2	3	5	2.0000	3.0000	5.0000
15	4.5000%	2	3	4	2.0000	3.0000	4.0000
16	1.8000%	2	3	3	2.0000	2.8333	3.1667
17	2.0250%	2	2	5	1.8333	2.1667	5.0000
18	1.6875%	2	2	4	1.8333	2.1667	4.0000
19	1.3500%	2	2	3	1.8333	2.1667	3.0000
20	0.3375%	2	2	2	1.7500	2.0000	2.2500
21	2.7000%	1	5	5	1.0000	4.8333	5.1667
22	4.5000%	1	4	5	1.0000	4.0000	5.0000
23	1.8750%	1	4	4	1.0000	3.8333	4.1667
24	3.6000%	1	3	5	1.0000	3.0000	5.0000
25	3.0000%	1	3	4	1.0000	3.0000	4.0000
26	1.2000%	1	3	3	1.0000	2.8333	3.1667
27	2.7000%	1	2	5	1.0000	2.0000	5.0000
28	2.2500%	1	2	4	1.0000	2.0000	4.0000
29	1.8000%	1	2	3	1.0000	2.0000	3.0000
30	0.6750%	1	2	2	1.0000	1.8333	2.1667
31	0.9000%	1	1	5	0.8333	1.1667	5.0000
32	0.7500%	1	1	4	0.8333	1.1667	4.0000
33	0.6000%	1	1	3	0.8333	1.1667	3.0000
34	0.4500%	1	1	2	0.8333	1.1667	2.0000
35	0.1000%	1	1	1	0.7500	1.0000	1.2500

Source: Adapted from file "CostList.txt" generated by the computer program "MS1File.bas".

Attachment D

Point Values and Equivalent Probabilities for the Weighting of Alternative Basic Parameters for the Scenario Analyses.

Cost-Spread Ratio (Low Cost/High Cost)	Point Values	Equivalent Probability
25.00%	1	12.5%
33.33%	1	12.5%
40.00%	2	25.0%
50.00%	2	25.0%
66.67%	2	25.0%

Fixed-Cost Portion Of Long-Run Cost	Point Values	Equivalent Probability
0.00%	1	12.5%
25.00%	2	25.0%
50.00%	2	25.0%
75.00%	2	25.0%
100.00%	1	12.5%

Transition Speed (Allowed Cost Level Jumps)	Point Values	Equivalent Probability
1.5	1	33.3%
2.5	1	33.3%
4.5	1	33.3%

Transition Length (Number of Years)	Point Values	Equivalent Probability
3	1	33.3%
5	1	33.3%
8	1	33.3%

Attachment E

Microsoft's Real Monopoly Revenues by Year
Discounted at 7% Rate per Year
Real 2001 dollars (in billions)

Year	Undiscounted Revenues	Discount Factor	Discounted Revenues
1995	3.003532	1.500731	4.507492
1996	3.727131	1.402552	5.227496
1997	5.035883	1.310796	6.601017
1998	6.454595	1.225043	7.907158
1999	7.693463	1.144900	8.808247
2000	8.186612	1.070000	8.759675
2001	7.204304	1.000000	7.204304
2002	9.142475	0.934579	8.544368
2003	10.588836	0.873439	9.248699
2004	12.001410	0.816298	9.796724
2005	13.364562	0.762895	10.195759
2006	14.670118	0.712986	10.459589
2007	15.915669	0.666342	10.605279
2008	17.102945	0.622750	10.650851
2009	18.236419	0.582009	10.613758
2010	19.322209	0.543934	10.509997
2011	20.367255	0.508349	10.353674
2012	21.378745	0.475093	10.156882
2013	22.363742	0.444012	9.929763
2014	23.328943	0.414964	9.680676
2015	24.280557	0.387817	9.416412
2016	25.224251	0.362446	9.142423
2017	26.165138	0.338734	8.863031
2018	27.107806	0.316574	8.581630
2019	28.056353	0.295864	8.300855
2020	29.014438	0.276508	8.022726
2021	29.985331	0.258419	7.748772
2022	30.971962	0.241513	7.480127
2023	31.976974	0.225713	7.217616
2024	33.002757	0.210947	6.961821
2025	34.051497	0.197146	6.713130

Source: Adapted from the "Rev_Disc.txt" file generated by the "MS6Summ.bas" program using RevStream=1 (Microsoft's Platform-only revenues) and the data in Attachment A.

Attachment F

Consumer Surplus & Profits
For Past (1995-2001) & Future (2002-2025) Time Intervals
Comparisons for Selected Remedies and Lawful Path

Time Interval	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
Aggregates for No Remedy Path:				
Past:	12.1839999	0.0000000	19.8218227	32.0058226
Future:	54.4862867	0.0000000	88.6422783	143.1285650
Total:	66.6702866	0.0000000	108.4641010	175.1343876
Aggregates for 100% Effective Conduct Remedy:				
Past:	12.1839999	0.0000000	19.8218227	32.0058226
Future:	84.8797426	24.7805427	37.2686222	146.9289075
Total:	97.0637425	24.7805427	57.0904448	178.9347301
Aggregates for 3-firm APM Structural Remedy:				
Past:	12.1839999	0.0000000	19.8218227	32.0058226
Future:	89.2098711	5.1158769	54.7189826	149.0447306
Total:	101.3938710	5.1158769	74.5408053	181.0505532
Aggregates for Lawful Path:				
Past:	16.3216726	2.6242527	13.0930073	32.0389326
Future:	89.5353459	31.5087389	28.2007864	149.2448711
Total:	105.8570185	34.1329915	41.2937937	181.2838037

Time Interval	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
Comparing: No Remedy minus LawfulPath:				
Past:	-4.1376727	-2.6242527	6.7288153	-0.0331101
Future:	-35.0490592	-31.5087389	60.4414920	-6.1163061
Total:	-39.1867319	-34.1329915	67.1703073	-6.1494161
Comparing: 100% Effective Conduct Remedy minus LawfulPath:				
Past:	-4.1376727	-2.6242527	6.7288153	-0.0331101
Future:	-4.6556032	-6.7281961	9.0678358	-2.3159636
Total:	-8.7932759	-9.3524488	15.7966511	-2.3490736
Comparing: 3-firm APM Structural Remedy minus LawfulPath:				
Past:	-4.1376727	-2.6242527	6.7288153	-0.0331101
Future:	-0.3254747	-26.3928620	26.5181963	-0.2001404
Total:	-4.4631474	-29.0171147	33.2470116	-0.2332505

Source: Adapted from output file "AGGRWTD8.txt" from Lundgren's six computer programs, where revstream=1 in "MS6Summ.bas".

Summary Output of Alternative Remedies for Microsoft.
Revenue Stream = Platforms only.
The value of z in the RPM scenarios is fixed as indicated.
Figures are in billions of real 2001 dollars (7% discount rate).
Figures are aggregated for the years 1995-2025.
Figures are a weighted average of all computed scenarios.

Total Aggregates for Alternative Remedies:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	66.6702866	0.0000000	108.4641010	175.1343876
20% Conduct:	72.7489789	4.9561086	98.1893712	175.8944587
40% Conduct:	78.8276711	9.9122172	87.9146414	176.6545298
60% Conduct:	84.9063629	14.8683262	77.6399093	177.4145984
80% Conduct:	90.9850527	19.8244345	67.3651770	178.1746642
100% Conduct:	97.0637425	24.7805427	57.0904448	178.9347301
2-Monopolies:	97.2443831	20.0995342	61.7667505	179.1106678
APM,2-firms:	97.6056643	10.7375170	71.1193619	179.4625432
APM,3-firms:	101.3938710	5.1158769	74.5408053	181.0505532
RPM,z=0.000:	97.6056643	10.7375170	71.1193619	179.4625432
RPM,z=0.100:	100.0494922	10.5198190	70.1711669	180.7404781
RPM,z=0.200:	101.7502572	10.2983154	70.2577918	182.3063644
RPM,z=0.300:	104.1852610	10.0963259	69.3834217	183.6650087
RPM,z=0.400:	104.3235767	9.9646972	70.6687936	184.9570675
RPM,z=0.500:	103.6437601	9.8490147	72.1858682	185.6786430
RPM,z=0.600:	100.2265622	9.7727729	75.5755929	185.5749280
RPM,z=0.700:	95.5628679	9.7447668	79.3962211	184.7038557
RPM,z=0.800:	89.8870594	9.7966324	83.5152687	183.1989605
RPM,z=0.900:	84.3841146	10.0564241	87.0578989	181.4984376
Lawful Path:	105.8570185	34.1329915	41.2937937	181.2838037

Comparing Remedies: Each remedy minus Lawful Path:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	-39.1867319	-34.1329915	67.1703073	-6.1494161
20% Conduct:	-33.1080412	-29.1768834	56.8955769	-5.3893477
40% Conduct:	-27.0293505	-24.2207753	46.6208465	-4.6292793
60% Conduct:	-20.9506588	-19.2646663	36.3461144	-3.8692107
80% Conduct:	-14.8719674	-14.3085576	26.0713827	-3.1091422
100% Conduct:	-8.7932759	-9.3524488	15.7966511	-2.3490736
2-Monopolies:	-8.6126354	-14.0334574	20.4729568	-2.1731359
APM,2-firms:	-8.2513542	-23.3954745	29.8255682	-1.8212605
APM,3-firms:	-4.4631474	-29.0171147	33.2470116	-0.2332505
RPM,z=0.000:	-8.2513542	-23.3954745	29.8255682	-1.8212605
RPM,z=0.100:	-5.8075263	-23.6131726	28.8773732	-0.5433256
RPM,z=0.200:	-4.1067613	-23.8346762	28.9639981	1.0225607
RPM,z=0.300:	-1.6717575	-24.0366656	28.0896281	2.3812049
RPM,z=0.400:	-1.5334418	-24.1682943	29.3749999	3.6732638
RPM,z=0.500:	-2.2132584	-24.2839768	30.8920745	4.3948392
RPM,z=0.600:	-5.6304563	-24.3602186	34.2817992	4.2911243
RPM,z=0.700:	-10.2941506	-24.3882248	38.1024274	3.4200520
RPM,z=0.800:	-15.9699591	-24.3363591	42.2214750	1.9151568
RPM,z=0.900:	-21.4729039	-24.0765674	45.7641052	0.2146339

Source: Adapted from output file "AGGCWTD8.txt" from Lundgren's six computer programs, where zbump=0.0 in "MS5TranR.bas" and revstream=1 in "MS6Summ.bas".

Summary Output of Alternative Remedies for Microsoft.
Revenue Stream = Platforms+Applications+Enterprise.
The value of z in the RPM scenarios is fixed as indicated.
Figures are in billions of real 2001 dollars (7% discount rate).
Figures are aggregated for the years 1995-2025.
Figures are a weighted average of all computed scenarios.

Total Aggregates for Alternative Remedies:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	193.2538881	0.0000000	314.3995669	507.6534549
20% Conduct:	211.1377498	14.6052416	284.1619090	509.9049004
40% Conduct:	229.0216116	29.2104832	253.9242510	512.1563459
60% Conduct:	246.9054718	43.8157260	223.6865862	514.4077840
80% Conduct:	264.7893265	58.4209665	193.4489212	516.6592142
100% Conduct:	282.6731812	73.0262070	163.2112562	518.9106443
2-Monopolies:	283.2511215	59.2815132	176.9034634	519.4360981
APM,2-firms:	284.4070022	31.7921257	204.2878778	520.4870056
APM,3-firms:	295.4965492	15.1737142	214.5194539	525.1897174
RPM,z=0.000:	284.4070022	31.7921257	204.2878778	520.4870056
RPM,z=0.100:	291.5721057	31.1496087	201.5156026	524.2373169
RPM,z=0.200:	296.5683886	30.4958291	201.7670167	528.8312345
RPM,z=0.300:	303.7018786	29.8995957	199.2199958	532.8214701
RPM,z=0.400:	304.1242099	29.5110516	202.9787968	536.6140583
RPM,z=0.500:	302.1262579	29.1695258	207.4368322	538.7326159
RPM,z=0.600:	292.1012489	28.9444187	217.3834949	538.4291624
RPM,z=0.700:	278.4197409	28.8617920	228.5928304	535.8743633
RPM,z=0.800:	261.7511839	29.0150366	240.6874245	531.4536450
RPM,z=0.900:	245.5925757	29.7822425	251.0843904	526.4592086
Lawful Path:	307.5699061	99.8842184	118.3262301	525.7803546

Comparing Remedies: Each remedy minus Lawful Path:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	-114.3160180	-99.8842184	196.0733368	-18.1268996
20% Conduct:	-96.4321609	-85.2789783	165.8356771	-15.8754620
40% Conduct:	-78.5483037	-70.6737381	135.5980175	-13.6240244
60% Conduct:	-60.6644435	-56.0684954	105.3603526	-11.3725863
80% Conduct:	-42.7805842	-41.4632534	75.1226894	-9.1211483
100% Conduct:	-24.8967249	-26.8580114	44.8850261	-6.8697102
2-Monopolies:	-24.3187846	-40.6027052	58.5772333	-6.3442565
APM,2-firms:	-23.1629040	-68.0920927	85.9616478	-5.2933489
APM,3-firms:	-12.0733569	-84.7105042	96.1932238	-0.5906372
RPM,z=0.000:	-23.1629040	-68.0920927	85.9616478	-5.2933489
RPM,z=0.100:	-15.9978005	-68.7346098	83.1893726	-1.5430376
RPM,z=0.200:	-11.0015175	-69.3883893	83.4407867	3.0508799
RPM,z=0.300:	-3.8680276	-69.9846227	80.8937657	7.0411155
RPM,z=0.400:	-3.4456962	-70.3731668	84.6525667	10.8337037
RPM,z=0.500:	-5.4436482	-70.7146926	89.1106021	12.9522613
RPM,z=0.600:	-15.4686572	-70.9397997	99.0572648	12.6488078
RPM,z=0.700:	-29.1501652	-71.0224264	110.2666003	10.0940087
RPM,z=0.800:	-45.8187223	-70.8691818	122.3611945	5.6732904
RPM,z=0.900:	-61.9773304	-70.1019759	132.7581603	0.6788540

Source: Adapted from output file "AGGCWTD8.txt" from Lundgren's six computer programs, where zbump=0.0 in "MS5TranR.bas" and revstream=4 in "MS6Summ.bas".

Summary Output of Alternative Remedies for Microsoft.
Revenue Stream = Platforms only.
The actual value of z in the RPM scenarios varies
between the indicated z and z-0.3.
Figures are in billions of real 2001 dollars (7% discount rate).
Figures are aggregated for the years 1995-2025.
Figures are a weighted average of all computed scenarios.

Total Aggregates for Alternative Remedies:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	66.6702866	0.0000000	108.4641010	175.1343876
20% Conduct:	72.7489789	4.9561086	98.1893712	175.8944587
40% Conduct:	78.8276711	9.9122172	87.9146414	176.6545298
60% Conduct:	84.9063629	14.8683262	77.6399093	177.4145984
80% Conduct:	90.9850527	19.8244345	67.3651770	178.1746642
100% Conduct:	97.0637425	24.7805427	57.0904448	178.9347301
2-Monopolies:	97.2443831	20.0995342	61.7667505	179.1106678
APM,2-firms:	97.6056643	10.7375170	71.1193619	179.4625432
APM,3-firms:	101.3938710	5.1158769	74.5408053	181.0505532
RPM,z=0.000:	97.6178004	10.7375170	71.0922687	179.4475861
RPM,z=0.100:	100.0792119	10.5200355	70.1142454	180.7134929
RPM,z=0.200:	102.4749421	10.3041911	69.0645211	181.8436543
RPM,z=0.300:	105.5117884	10.1207266	67.4459993	183.0785144
RPM,z=0.400:	107.9097123	10.0131838	66.0581453	183.9810414
RPM,z=0.500:	109.3712929	9.9176811	65.7228881	185.0118620
RPM,z=0.600:	110.6707337	9.8362632	65.0862391	185.5932361
RPM,z=0.700:	109.3070756	9.7864170	67.0511494	186.1446421
RPM,z=0.800:	106.7436886	9.7342133	69.7748845	186.2527864
RPM,z=0.900:	101.4498564	9.7168714	74.5723925	185.7391203
Lawful Path:	105.8570185	34.1329915	41.2937937	181.2838037

Comparing Remedies: Each remedy minus Lawful Path:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	-39.1867319	-34.1329915	67.1703073	-6.1494161
20% Conduct:	-33.1080412	-29.1768834	56.8955769	-5.3893477
40% Conduct:	-27.0293505	-24.2207753	46.6208465	-4.6292793
60% Conduct:	-20.9506588	-19.2646663	36.3461144	-3.8692107
80% Conduct:	-14.8719674	-14.3085576	26.0713827	-3.1091422
100% Conduct:	-8.7932759	-9.3524488	15.7966511	-2.3490736
2-Monopolies:	-8.6126354	-14.0334574	20.4729568	-2.1731359
APM,2-firms:	-8.2513542	-23.3954745	29.8255682	-1.8212605
APM,3-firms:	-4.4631474	-29.0171147	33.2470116	-0.2332505
RPM,z=0.000:	-8.2392181	-23.3954745	29.7984750	-1.8362176
RPM,z=0.100:	-5.7778066	-23.6129560	28.8204517	-0.5703108
RPM,z=0.200:	-3.3820764	-23.8288004	27.7707274	0.5598506
RPM,z=0.300:	-0.3452300	-24.0122649	26.1522056	1.7947106
RPM,z=0.400:	2.0526938	-24.1198077	24.7643516	2.6972377
RPM,z=0.500:	3.5142744	-24.2153105	24.4290944	3.7280583
RPM,z=0.600:	4.8137152	-24.2967283	23.7924455	4.3094323
RPM,z=0.700:	3.4500571	-24.3465745	25.7573557	4.8608383
RPM,z=0.800:	0.8866701	-24.3987783	28.4810908	4.9689827
RPM,z=0.900:	-4.4071621	-24.4161201	33.2785988	4.4553166

Source: Adapted from output file "AGGCWTD8.txt" from Lundgren's six computer programs, where zbump=0.3 in "MS5TranR.bas" and revstream=1 in "MS6Summ.bas".

Attachment J

Summary Output of Alternative Remedies for Microsoft.
Revenue Stream = Platforms+Applications+Enterprise.
The actual value of z in the RPM scenarios varies
between the indicated z and z-0.3.
Figures are in billions of real 2001 dollars (7% discount rate).
Figures are aggregated for the years 1995-2025.
Figures are a weighted average of all computed scenarios.

Total Aggregates for Alternative Remedies:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	193.2538881	0.0000000	314.3995669	507.6534549
20% Conduct:	211.1377498	14.6052416	284.1619090	509.9049004
40% Conduct:	229.0216116	29.2104832	253.9242510	512.1563459
60% Conduct:	246.9054718	43.8157260	223.6865862	514.4077840
80% Conduct:	264.7893265	58.4209665	193.4489212	516.6592142
100% Conduct:	282.6731812	73.0262070	163.2112562	518.9106443
2-Monopolies:	283.2511215	59.2815132	176.9034634	519.4360981
APM,2-firms:	284.4070022	31.7921257	204.2878778	520.4870056
APM,3-firms:	295.4965492	15.1737142	214.5194539	525.1897174
RPM,z=0.000:	284.4427827	31.7921257	204.2079998	520.4429082
RPM,z=0.100:	291.6597578	31.1502485	201.3477301	524.1577365
RPM,z=0.200:	298.6895994	30.5131724	198.2740901	527.4768619
RPM,z=0.300:	307.5955291	29.9716086	193.5330233	531.1001609
RPM,z=0.400:	314.6358118	29.6541255	189.4628042	533.7527415
RPM,z=0.500:	318.9329183	29.3722039	188.4709348	536.7760570
RPM,z=0.600:	322.7383754	29.1318953	186.6145056	538.4847762
RPM,z=0.700:	318.7571699	28.9848037	192.3609534	540.1029270
RPM,z=0.800:	311.2289728	28.8306843	200.3597163	540.4193734
RPM,z=0.900:	295.6928896	28.7794391	214.4391657	538.9114945
Lawful Path:	307.5699061	99.8842184	118.3262301	525.7803546

Comparing Remedies: Each remedy minus Lawful Path:

Remedy	Consumer Surplus	Competitor Profits	Microsoft Profits	Total Surplus
No-Remedy:	-114.3160180	-99.8842184	196.0733368	-18.1268996
20% Conduct:	-96.4321609	-85.2789783	165.8356771	-15.8754620
40% Conduct:	-78.5483037	-70.6737381	135.5980175	-13.6240244
60% Conduct:	-60.6644435	-56.0684954	105.3603526	-11.3725863
80% Conduct:	-42.7805842	-41.4632534	75.1226894	-9.1211483
100% Conduct:	-24.8967249	-26.8580114	44.8850261	-6.8697102
2-Monopolies:	-24.3187846	-40.6027052	58.5772333	-6.3442565
APM,2-firms:	-23.1629040	-68.0920927	85.9616478	-5.2933489
APM,3-firms:	-12.0733569	-84.7105042	96.1932238	-0.5906372
RPM,z=0.000:	-23.1271235	-68.0920927	85.8817698	-5.3374464
RPM,z=0.100:	-15.9101483	-68.7339699	83.0215001	-1.6226181
RPM,z=0.200:	-8.8803067	-69.3710460	79.9478600	1.6965073
RPM,z=0.300:	0.0256230	-69.9126099	75.2067932	5.3198063
RPM,z=0.400:	7.0659057	-70.2300929	71.1365742	7.9723870
RPM,z=0.500:	11.3630122	-70.5120145	70.1447047	10.9957024
RPM,z=0.600:	15.1684692	-70.7523231	68.2882755	12.7044217
RPM,z=0.700:	11.1872638	-70.8994147	74.0347234	14.3225724
RPM,z=0.800:	3.6590666	-71.0535341	82.0334862	14.6390188
RPM,z=0.900:	-11.8770165	-71.1047793	96.1129357	13.1311399

Source: Adapted from output file "AGGCWTD8.txt" from Lundgren's six computer programs, where zbump=0.3 in "MS5TranR.bas" and revstream=4 in "MS6Summ.bas".

Documentation for BASIC Programs to Simulate
Antitrust Remedies for Microsoft Case.

This document, "MS_Sim_Doc.txt", simply describes and documents
six programs for the Microsoft antitrust remedy simulations.
These six programs are named:

```
MS1File.bas      (0.2 seconds)
MS2ProbA.bas     (10.3 minutes)
MS3ProbR.bas     (18.1 minutes)
MS4TranA.bas     ( 1.6 minutes)
MS5TranR.bas     (24.9 minutes)
MS6Summ.bas      ( 1.7 minutes)
```

The programs should be run in the order indicated, since files generated by one program are used by subsequent programs. The running times are approximate, based on the running times for a 1.6 GHz home computer. The programs were coded and run in Microsoft QuickBASIC. The programs may require some recoding, if it is desired to run them in other versions of the BASIC computer language.

Below is a summary description of what each program does. Each program has its own more detailed description.

Program MS1File.bas:

This program generates files needed by subsequent programs.
The program generates the "COSTLIST.txt" file,
which details the assumed cost levels for each scenario.
For 3 firms and 5 levels of cost, 35 cost scenarios are generated.
The program also generates the "Ordering.txt" and "OrderRPM.txt" files.
These files generate the permutations by which the ranking of firms
can be reordered. For 3 firms, there are 6 permutations.
"OrderRPM.txt" allows the "MS3ProbR.bas" program to track the rankings
of two Microsoft successor firms simultaneously.

Program MS2ProbA.bas:

This program computes the probabilites associated with each
scenario, as the industry transitions from a particular starting point,
and gradually converges towards a long-run stochastic equilibrium.
This program assigns probabilities for equilibria consisting only
of Absolute Profit Maximizing ("APM") firms.
The starting point varies by the number of Microsoft firms (msfirms)
in period zero. If msfirms=1, Microsoft starts as a monopoly.
If msfirms=2, Microsoft is split into two firms.

If msfirms=3, Microsoft is split into three firms.
The program uses three different speeds (speed=1,2,3) for the transition.
Probability files are outputted for each msfirms=1,2,3 starting point,
and each speed=1,2,3 for the transition speed.

Program MS3ProbR.bas:
This program is similar to "MS2ProbA.bas", since it computes probabilites associated with each scenario, as the industry transitions
from a particular starting point, and gradually converges towards a
long-run stochastic equilibrium. This program differs from "MS2ProbA.bas",
because it assigns probabilities for equilibria consisting
of two Relative Profit Maximizing ("RPM") firms, along with such APM
firms as may be involved in the transitions. The equilibria automatically
convert to APM equilibria if one or both RPM firms exits the industry.
The program uses three different speeds (speed=1,2,3) for the transition.
Probability files are outputted for the one starting point (msfirms=2),
and each speed=1,2,3 for the transition speed. This program is more
complex than "MS2ProbA.bas" because it must simultaneously track the
rankings of two Microsoft-successor firms simultaneously.

Program MS4TranA.bas:
This program uses the probability data computed by "MS2ProbA.bas"
to compute Consumer Surplus and Profits for both Microsoft and
Microsoft's competitors. These are determined for transition period
zero (iter=0) under the assumption that Microsoft has no competitors
in period zero. In transition periods one through ten, Microsoft is
assumed to have (at least potentially) one or more competitors.
This program only calculates APM equilibria.
The 225 outputted transition (TRAN....txt) files are computed
for three speeds of transition (speed=1,2,3),
five cost ratios for short-run cost (cratio=1,2,3,4,5),
five assumptions concerning the portion of long-run costs
allocated to fixed costs (port=0,1,2,3,4),
and three starting points (msfirms=1,2,3).

Program MS5TranR.bas:

This program uses the probability data computed by "MS3ProbR.bas" to compute Consumer Surplus and Profits for both Microsoft and Microsoft's competitors. These are determined for transition period zero (iter=0) under the assumption that Microsoft has no competitors in period zero. In transition periods one through ten, Microsoft is assumed to have (at least potentially) one or more competitors. This program calculates both RPM and APM equilibria.

The 750 outputted transition (TRPM....txt) files are computed for three speeds of transition (speed=1,2,3), five cost ratios for short-run cost (cratio=1,2,3,4,5), five assumptions concerning the portion of long-run costs allocated to fixed costs (port=0,1,2,3,4), and ten starting points (z = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9). The starting point always has Microsoft divided into two RPM firms, where the goal functions for the two firms are:

```
Goal1 = Profit1 - z * Profit2
Goal2 = Profit2 - z * Profit1
```

An additional feature of the program allows the value of z to change in response to circumstances. If zbump=0.0, then z is fixed, and does not change in response to circumstances. If zbump > 0, then z changes in response to circumstances. In the program, z responds to the circumstance that one of the RPM firms is not producing, because it is achieving negative absolute profit. In this circumstance, the program automatically "bumps down" the value of z for both RPM firms by the amount of zbump. For example, if z=0.7 and zbump=0.4, then if one or both RPM firms would shut down, then the value of z is automatically bumped down to z=0.3. In many circumstances, this allows both RPM firms to continue producing.

Program MS6Summ.bas:

This program computes and summarizes the data produced by prior programs, including both "MS4TranA.bas" and "MS5TranR.bas". The program produces data summarized for particular scenarios in files marked "AGGC....txt", "AGGR....txt", and "YEAR....txt". The "AGGC....txt" files (which are most user friendly) summarize all past and future data, appropriately discounted, into a single set of figures which may be compared across remedy proposals. The "AGGR....txt" files categorize the aggregate data into

past and future amounts of consumer surplus, profits, and total surplus for each remedy proposal, and how these amounts compare with the same amounts along the lawful path.
The "YEAR....txt" files (which are least user friendly) output the calculated amounts, by year, for each remedy proposal and the lawful path.

Attachment L.

```
'BASIC Program "MS1File.bas".

'Program Number 1 in a series of six programs
'designed to simulate alternative antitrust
'remedies for the Microsoft software industry.

'Copyright, January 23, 2002, Carl Lundgren.

'   This program, "MS1File.bas", generates files
'needed by the subsequent computer programs
'for the Microsoft antitrust remedy simulations.
'This program generates the "COSTLIST.txt" file,
'which details the assumed cost levels for each scenario.
'For 3 firms and 5 levels of cost, 35 cost scenarios are
generated.
'   This program also generates the "Ordering.txt"
'and "OrderRPM.txt" files.  These files generate the
'permutations by which the ranking of firms can be reordered.
'For 3 firms, there are 6 permutations.
'The file "OrderRPM.txt" allows the "MS3ProbR.bas" program to
track
'the rankings of two Microsoft successor firms simultaneously.

DEFDBL A-Z
DIM broadscen(1023), class(5), cost(5)
DIM weight(50), newclass1(50), newclass2(50), newclass3(50)
DIM newclass4(50), newclass5(50)
DIM c1(50), c2(50), c3(50), c4(50), c5(50)
DIM pv(50, 3), finprob(50)
DIM new(5), ORDER(6, 3), ORDERRPM(6, 15)
timex = TIMER
CLS

GOSUB GENERATE:
GOSUB COLLAPSE:
GOSUB COSTIT:
GOSUB PVASSIGN:
GOSUB FINALPROB:
GOSUB PRINTCOST:
GOSUB ORDER:
GOSUB PRINTORDER:
GOSUB PRINTORDERRPM:
```

```
PRINT TIMER - timex
END


GENERATE:
'Submodule to generate possible scenarios.
FOR scennum = 0 TO 215
   broadscen(scennum) = 0
NEXT scennum
FOR firm1 = 1 TO 5
FOR firm2 = 1 TO 5
FOR firm3 = 1 TO 5
   GOSUB CLASSIFY:
NEXT firm3
NEXT firm2
NEXT firm1
RETURN
'*****END of Generate Submodule*****


CLASSIFY:
'Submodule of Generate submodule
'   to classify the generated scenarios.
class(1) = 0
class(2) = 0
class(3) = 0
class(4) = 0
class(5) = 0
class(firm1) = class(firm1) + 1
class(firm2) = class(firm2) + 1
class(firm3) = class(firm3) + 1
scennum = 256 * class(1) + 64 * class(2) + 16 * class(3) + 4 *
class(4) + class(5)
broadscen(scennum) = broadscen(scennum) + 1
RETURN
'*****END of Classify Submodule*****


COLLAPSE:
'Submodule to collapse the number of scenarios
'   to a more manageable number.
newnum = 0
FOR class1 = 0 TO 3
FOR class2 = 0 TO 3
```

```
FOR class3 = 0 TO 3
FOR class4 = 0 TO 3
FOR class5 = 0 TO 3
   scennum = 256 * class1 + 64 * class2 + 16 * class3 + 4 *
class4 + class5
   broadnum = broadscen(scennum)
   IF broadnum > 0 THEN
      newnum = newnum + 1
      weight(newnum) = broadnum
      newclass1(newnum) = class1
      newclass2(newnum) = class2
      newclass3(newnum) = class3
      newclass4(newnum) = class4
      newclass5(newnum) = class5
   END IF
NEXT class5
NEXT class4
NEXT class3
NEXT class2
NEXT class1
newtot = newnum
RETURN
'*****END of Collapse Submodule*****


COSTIT:
'Submodule to assign cost levels to firms,
'   with lowest-cost firms ordered first.
FOR scen = 1 TO newtot
   n1 = newclass1(scen)
   n2 = newclass2(scen) + n1
   n3 = newclass3(scen) + n2
   n4 = newclass4(scen) + n3
   FOR n = 1 TO n1
      cost(n) = 1
   NEXT n
   FOR n = n1 + 1 TO n2
      cost(n) = 2
   NEXT n
   FOR n = n2 + 1 TO n3
      cost(n) = 3
   NEXT n
   FOR n = n3 + 1 TO n4
      cost(n) = 4
```

```
   NEXT n
   FOR n = n4 + 1 TO 3
      cost(n) = 5
   NEXT n
   c1(scen) = cost(1)
   c2(scen) = cost(2)
   c3(scen) = cost(3)
NEXT scen
RETURN
'*****END of Costit Submodule*****


PVASSIGN:
'Submodule to assign point values for firm cost levels,
'   with lowest-cost firms ordered first.
'   The point values are 60 times the cost level,
'   with some adjustment in point values, when
'   two or more firms share the same cost level.
FOR scen = 1 TO newtot
   pv(scen, 1) = c1(scen) * 60
   pv(scen, 2) = c2(scen) * 60
   pv(scen, 3) = c3(scen) * 60
NEXT scen
FOR scen = 1 TO newtot
'   n1 = newclass1(scen)
'   n2 = newclass2(scen) + n1
'   n3 = newclass3(scen) + n2
'   n4 = newclass4(scen) + n3
'Assign point values to level one costs.
   ns = 0
   nc = newclass1(scen)
   IF nc = 2 THEN
      pv(scen, ns + 1) = pv(scen, ns + 1) - 10
      pv(scen, ns + 2) = pv(scen, ns + 2) + 10
      END IF
   IF nc = 3 THEN
      pv(scen, ns + 1) = pv(scen, ns + 1) - 15
      pv(scen, ns + 3) = pv(scen, ns + 3) + 15
      END IF
'Assign point values to level two costs.
   ns = ns + nc
   nc = newclass2(scen)
   IF nc = 2 THEN
      pv(scen, ns + 1) = pv(scen, ns + 1) - 10
```

```
         pv(scen, ns + 2) = pv(scen, ns + 2) + 10
         END IF
      IF nc = 3 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 15
         pv(scen, ns + 3) = pv(scen, ns + 3) + 15
         END IF
'Assign point values to level three costs.
      ns = ns + nc
      nc = newclass3(scen)
      IF nc = 2 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 10
         pv(scen, ns + 2) = pv(scen, ns + 2) + 10
         END IF
      IF nc = 3 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 15
         pv(scen, ns + 3) = pv(scen, ns + 3) + 15
         END IF
'Assign point values to level four costs.
      ns = ns + nc
      nc = newclass4(scen)
      IF nc = 2 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 10
         pv(scen, ns + 2) = pv(scen, ns + 2) + 10
         END IF
      IF nc = 3 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 15
         pv(scen, ns + 3) = pv(scen, ns + 3) + 15
         END IF
'Assign point values to level five costs.
      ns = ns + nc
      nc = newclass5(scen)
      IF nc = 2 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 10
         pv(scen, ns + 2) = pv(scen, ns + 2) + 10
         END IF
      IF nc = 3 THEN
         pv(scen, ns + 1) = pv(scen, ns + 1) - 15
         pv(scen, ns + 3) = pv(scen, ns + 3) + 15
         END IF
NEXT scen
RETURN
'*****END of PVassign Submodule*****
```

```
FINALPROB:
'This submodule computes the final probability
'   for each scenario--the probability toward which
'   each scenario tends to converge over the long run.
'cost(s,f) = short-run marginal cost of firm f in scenario s.
'finprob(s) = final probability assumed for scenario s.
'weight(s) = number of permutations of scenario s.
'finprob is computed as weight(s) * assumed probabilities
'   for each cost level:
'   Prob(cost level one) = 10% (low cost)
'   Prob(cost level two) = 15% (low-middle cost)
'   Prob(cost level three) = 20% (middle cost)
'   Prob(cost level four) = 25% (middle-high cost)
'   Prob(cost level five) = 30% (high cost)
FOR scen = 1 TO newtot
   fprob = weight(scen)
   L1 = newclass1(scen)
   L2 = newclass2(scen)
   L3 = newclass3(scen)
   L4 = newclass4(scen)
   L5 = newclass5(scen)
     IF L1 > 0 THEN fprob = fprob * .1#
     IF L1 > 1 THEN fprob = fprob * .1#
     IF L1 > 2 THEN fprob = fprob * .1#
     IF L2 > 0 THEN fprob = fprob * .15#
     IF L2 > 1 THEN fprob = fprob * .15#
     IF L2 > 2 THEN fprob = fprob * .15#
     IF L3 > 0 THEN fprob = fprob * .2#
     IF L3 > 1 THEN fprob = fprob * .2#
     IF L3 > 2 THEN fprob = fprob * .2#
     IF L4 > 0 THEN fprob = fprob * .25#
     IF L4 > 1 THEN fprob = fprob * .25#
     IF L4 > 2 THEN fprob = fprob * .25#
     IF L5 > 0 THEN fprob = fprob * .3#
     IF L5 > 1 THEN fprob = fprob * .3#
     IF L5 > 2 THEN fprob = fprob * .3#
   finprob(scen) = fprob
NEXT scen
RETURN
'***** END OF FinalProb SUBMODULE *****


PRINTCOST:
'Submodule to print out the collapsed scenarios
```

```
'   and the ordered cost assignments
'   as part of file "CostList.txt".
cost$ = "c:\basic\ms_sim\costlist.txt" 'Output cost list
OPEN cost$ FOR OUTPUT AS #1
PRINT #1, "Scen"; " L1 L2 L3 L4 L5"; "  Wgt"; " Fin-Prob";
PRINT #1, "   C1 C2 C3"; "   PV1 PV2 PV3"
FOR scennum = 1 TO newtot
   PRINT #1, USING "##"; scennum;
   PRINT #1, "  ";
   PRINT #1, USING "###"; newclass1(scennum);
   PRINT #1, USING "###"; newclass2(scennum);
   PRINT #1, USING "###"; newclass3(scennum);
   PRINT #1, USING "###"; newclass4(scennum);
   PRINT #1, USING "###"; newclass5(scennum);
   PRINT #1, USING "#####"; weight(scennum);
   PRINT #1, USING "###.######"; finprob(scennum);
   PRINT #1, " ";
   PRINT #1, USING "###"; c1(scennum);
   PRINT #1, USING "###"; c2(scennum);
   PRINT #1, USING "###"; c3(scennum);
   PRINT #1, "  ";
   PRINT #1, USING "####"; pv(scennum, 1);
   PRINT #1, USING "####"; pv(scennum, 2);
   PRINT #1, USING "####"; pv(scennum, 3);
   PRINT #1,
NEXT scennum
CLOSE #1
RETURN
'*****END of PrintCost Submodule*****


ORDER:
'Submodule to compute all possible orderings
'   of three firms (six permutations total).
ordernum = 0
FOR o3 = 5 TO 1 STEP -1
FOR o2 = 5 TO 1 STEP -1
FOR o1 = 5 TO 1 STEP -1
   GOSUB TESTORDER:
NEXT o1
NEXT o2
NEXT o3
ordertot = ordernum
RETURN
```

```
'*****END of Order Submodule*****


TESTORDER:
'Submodule of Order submodule to test
'   whether proposed ordering is acceptable.
IF o1 + o2 + o3 <> 6 THEN RETURN
IF o1 * o2 * o3 <> 6 THEN RETURN
ordernum = ordernum + 1
ORDER(ordernum, 1) = o1
ORDER(ordernum, 2) = o2
ORDER(ordernum, 3) = o3
GOSUB ORDERRPM:
RETURN
'*****END of Order Submodule*****


ORDERRPM:
'Submodule to provide ordering information
'   to track location of two MS firms among five firms,
'   for purpose of determining costs of such
'   two firms for calculating RPM remedy.
'   There are six basic permutations of three firms,
'   among which the rankings of two firms must be
'   tracked simultaneously.
new(0) = 0
FOR old = 1 TO 3
   new(old) = ORDER(ordernum, old)
NEXT old
FOR old1 = 0 TO 3
FOR old2 = 0 TO 3
   oldnum = old1 * 4 + old2
   new1 = new(old1)
   new2 = new(old2)
   newnum = new1 * 4 + new2
   ORDERRPM(ordernum, oldnum) = newnum
NEXT old2
NEXT old1
RETURN
'*****END of OrderRPM Submodule*****


PRINTORDER:
'Submodule to print out the 6 permutations
```

```
'   in which 3 firms can be ordered.
'   Printing is to the file "Ordering.txt".
ORDER$ = "c:\basic\ms_sim\ordering.txt" 'Output ordering list
OPEN ORDER$ FOR OUTPUT AS #1
PRINT #1, "Onum"; " o1 o2 o3"
FOR ordernum = 1 TO ordertot
   PRINT #1, USING "###"; ordernum;
   PRINT #1, " ";
   PRINT #1, USING "###"; ORDER(ordernum, 1);
   PRINT #1, USING "###"; ORDER(ordernum, 2);
   PRINT #1, USING "###"; ORDER(ordernum, 3);
   PRINT #1,
NEXT ordernum
CLOSE #1
RETURN
'*****END of PrintOrder Submodule*****


PRINTORDERRPM:
'Submodule to print out the 6 permutations
'   in which 3 firms can be ordered,
'   with further information to track two firms
'   simultaneously, for further use in later
'   programs to calculate the effects of RPM firms.
'   Printing is to the file "OrderRPM.txt".
ORDERRPM$ = "c:\basic\ms_sim\orderrpm.txt" 'Output RPM ordering
list
OPEN ORDERRPM$ FOR OUTPUT AS #1
PRINT #1, "Onum"; " o00 o01 o02 o03";
PRINT #1, " o10 o11 o12 o13";
PRINT #1, " o20 o21 o22 o23";
PRINT #1, " o30 o31 o32 o33";
PRINT #1,
FOR ordernum = 1 TO ordertot
   PRINT #1, USING "###"; ordernum;
   PRINT #1, " ";
FOR oldnum = 0 TO 15
   PRINT #1, USING "####"; ORDERRPM(ordernum, oldnum);
NEXT oldnum
PRINT #1,
NEXT ordernum
CLOSE #1
RETURN
'*****END of PrintOrderRPM Submodule*****
```

```
'**********END OF Program "MS1File.bas".**********
```

Attachment M.

```
'BASIC Program "MS2ProbA.bas".

'Program Number 2 in a series of six programs
'designed to simulate alternative antitrust
'remedies for the Microsoft software industry.

'Copyright, January 23, 2002, Carl Lundgren.

'   This program, "MS2ProbA.bas", computes the probabilites
'associated with each scenario, as the industry transitions
'from a particular starting point, and gradually converges
'towards a long-run stochastic equilibrium.
'This program assigns probabilities for equilibria consisting
'only of Absolute Profit Maximizing ("APM") firms.
'The starting point varies by the number of
'Microsoft firms (msfirms) in period zero:
'     If msfirms=1, Microsoft starts as a monopoly.
'     If msfirms=2, Microsoft is split into two firms.
'     If msfirms=3, Microsoft is split into three firms.
'The program uses three different speeds (speed=1,2,3) for the
transition.
'Probability files are outputted for each starting point
(msfirms=1,2,3),
'and for each transition speed (speed=1,2,3).

'    The parameters controlling the transition speed
' (pvmax in submodule InitProb10) are supplied by the user.
' The program reads in 35 possible cost structures
' for the industry, each with 3 firms.
' The program assigns probabilities for each scenario,
' and for whether a Microsoft firm (either Microsoft or
' a successor to Microsoft after divestiture) is ranked
' as firm 1, 2, or 3, or is firm 0 (with zero market share).

DEFDBL A-Z
DIM pvtot0(35, 3), pvtot1(35, 3), pvtot2(35, 3)
DIM pvtot3(35, 3)
DIM prob2(35, 3)
DIM diff(3, 3)
DIM finprob(35)
```

```
'CONTROL MODULE
CLS
timex = TIMER
'This section calls the main module 3 times.
'This control module chooses speed for cost shifts:
   'speed = 1 'Slow speed for cost shifts.
   'speed = 2 'Moderate speed for cost shifts.
   'speed = 3 'High speed for cost shifts.
FOR speed = 1 TO 3
   GOSUB MAINMODULE:
NEXT speed
PRINT TIMER - timex
END


MAINMODULE:
GOSUB FILENAMES: 'Assign file names to input/output files.

PRINT "Computing transition weights (deviation):"
GOSUB INITIALIZE0: 'Initialize transition weights.
endcomp = 0
FOR iter = 1 TO 100
   GOSUB TRANSCOMP: 'Iterate transition weights.
   IF endcomp = 1 THEN 99
NEXT iter
99 GOSUB PRINTPROBT: 'Print last computed transition weights.
CLOSE #2

PRINT "Computing transitions from MS=1 APM firm:"
msfirms = 1
iter = 0
GOSUB INITIALIZE1: 'Microsoft starts as monopoly.
iter = 1
GOSUB TRANSIT0:
GOSUB PRINTPROB:
FOR iter = 2 TO 10
   GOSUB TRANSFERPROB:
   GOSUB TRANSIT1:
   GOSUB PRINTPROB:
NEXT iter
CLOSE #2

PRINT "Computing transitions from MS=2 APM firms:"
msfirms = 2
```

```
iter = 0
GOSUB INITIALIZE2: 'Microsoft split into 2 firms.
iter = 1
GOSUB TRANSIT0:
GOSUB PRINTPROB:
FOR iter = 2 TO 10
   GOSUB TRANSFERPROB:
   GOSUB TRANSIT1:
   GOSUB PRINTPROB:
NEXT iter
CLOSE #2

PRINT "Computing transitions from MS=3 APM firms:"
msfirms = 3
iter = 0
GOSUB INITIALIZE3: 'Microsoft split into 3 firms.
iter = 1
GOSUB TRANSIT0:
GOSUB PRINTPROB:
FOR iter = 2 TO 10
   GOSUB TRANSFERPROB:
   GOSUB TRANSIT1:
   GOSUB PRINTPROB:
NEXT iter
CLOSE #2

CLOSE
RETURN
'***** END OF MAIN MODULE *****


FILENAMES:
cost$ = "c:\basic\ms_sim\costlist.txt" 'Input scenario costs
order$ = "c:\basic\ms_sim\ordering.txt" 'Input firm re-orderings
prob0$ = "c:\basic\ms_sim\out\prob00.txt" 'Output iwgt
convergence
prob1$ = "c:\basic\ms_sim\out\prob10.txt" 'Output 1-firm APM
transition probs
prob2$ = "c:\basic\ms_sim\out\prob20.txt" 'Output 2-firm APM
transition probs
prob3$ = "c:\basic\ms_sim\out\prob30.txt" 'Output 3-firm APM
transition probs
IF speed = 1 THEN sp$ = "1"
IF speed = 2 THEN sp$ = "2"
```

```
IF speed = 3 THEN sp$ = "3"
replace$ = sp$
MID$(prob0$, 26, 1) = replace$
MID$(prob1$, 26, 1) = replace$
MID$(prob2$, 26, 1) = replace$
MID$(prob3$, 26, 1) = replace$
RETURN
'***** END OF FileNames SUBMODULE *****


INITIALIZE0:
'Submodule to find transition weights.
GOSUB SCENREAD: 'Read in scenario list.
GOSUB ORDERREAD: 'Read in ordering list.
GOSUB INITPROB10:
iter = 0
GOSUB TRANSCOMP: 'Computes transition weights to scenarios.
OPEN prob0$ FOR OUTPUT AS #2
RETURN
'***** END OF Initialize0 SUBMODULE *****


INITPROB10:
'This submodule sets the prob1 variables to zero,
'   and then sets initial values for non-zero prob1.
DIM prob1(35, 3), cost(35, 3), herf(35)
DIM iwgt(35), iwgt0(35)
'****User supplies pvmax, which controls transition speed.****
IF speed = 1 THEN pvmax = 1.5 'Slow speed for cost shifts.
IF speed = 2 THEN pvmax = 2.5 'Moderate speed for cost shifts.
IF speed = 3 THEN pvmax = 4.5 'High speed for cost shifts.
FOR scen1 = 0 TO 35
FOR firm1 = 0 TO 3
   prob1(scen1, firm1) = 0
NEXT firm1
NEXT scen1
'This section sets initial values  to reflect
'   distribution of final probabilites.
FOR scen1 = 1 TO 35
   prob1(scen1, 0) = finprob(scen1)
   iwgt(scen1) = finprob(scen1)
NEXT scen1
RETURN
'***** END OF InitProb10 SUBMODULE *****
```

```
TRANSCOMP:
'Submodule to compute transition weights.
'   Transitions are from any scenario (scen1)
'   to any same or different scenario (scen2).
'   Goal is to find transition weights (iwgt) such that
'   if prob1 is set at final probabilities,
'   then computed prob2 also reflects final probabilities.
'   The transition weights are iteratively adjusted,
'   until there is convergence.  Such convergence
'   means that the long-run distribution of scenarios
'   will reflect the final probabilities selected.
GOSUB INITPROB2: 'Initialize prob2 variables.
PRINT speed; iter; "*";
FOR scen1 = 1 TO 35
      PRINT ".";
   iprob0 = prob1(scen1, 0)
   iprob1 = prob1(scen1, 1)
   iprob2 = prob1(scen1, 2)
   iprob3 = prob1(scen1, 3)
FOR scen2 = 1 TO 35
   GOSUB PVADD:
NEXT scen2
   GOSUB PVADJUST:
NEXT scen1
devtot = 0
itotal = 0
FOR scen = 1 TO 35
   iwgt0(scen) = iwgt(scen)
   iwgt(scen) = iwgt(scen) * prob1(scen, 0) / prob2(scen, 0)
   itotal = itotal + iwgt(scen)
   dev = prob1(scen, 0) - prob2(scen, 0)
   devtot = devtot + ABS(dev)
NEXT scen
FOR scen = 1 TO 35
   iwgt(scen) = iwgt(scen) / itotal
NEXT scen
IF devtot < .000001 THEN endcomp = 1
PRINT USING "#.########"; devtot
RETURN
'This submodule finds transition weights (iwgt) to each scenario,
'   that cause convergence to the assumed final probabilities
(FinProb)
'   attached to the various possible market outcomes
```

```
'   in a very long-run stochastic equilibrium.
'***** END OF TransComp SUBMODULE *****


INITIALIZE1:
'Submodule to initialize Microsoft starts as monopoly.
GOSUB INITPROB11:
OPEN prob1$ FOR OUTPUT AS #2
GOSUB PRINTPROB0:
RETURN
'***** END OF Initialize1 SUBMODULE *****


INITPROB11:
'This submodule of Initialize1 sets the prob1 variables to zero,
'   and then sets initial values for non-zero prob1.
FOR scen1 = 0 TO 35
FOR firm1 = 0 TO 3
   prob1(scen1, firm1) = 0
NEXT firm1
NEXT scen1
'This section sets initial scenario to
'   Microsoft is a monopoly,
'   Scenario 5, Cost levels 3(MS), 5(comp), 5(comp).
scen0 = 5
prob1(0, 1) = 1
FOR firm = 1 TO 3
   cost(0, firm) = cost(scen0, firm)
NEXT firm
RETURN
'***** END OF InitProb11 SUBMODULE *****


INITIALIZE2:
'Submodule to initialize splitting Microsoft into two firms.
GOSUB INITPROB12:
OPEN prob2$ FOR OUTPUT AS #2
GOSUB PRINTPROB0:
RETURN
'***** END OF Initialize2 SUBMODULE *****


INITPROB12:
'This submodule sets the prob1 variables to zero,
```

```
'   and then sets initial values for non-zero prob1.
FOR scen1 = 0 TO 35
FOR firm1 = 0 TO 3
   prob1(scen1, firm1) = 0
NEXT firm1
NEXT scen1
'This section sets initial scenario to
'   Microsoft is split into two equal-sized firms,
'   Scenario 6, Cost levels 3(MS-1), 4(MS-2), 5(comp).
scen0 = 6
prob1(0, 1) = 1# / 2#
prob1(0, 2) = 1# / 2#
FOR firm = 1 TO 3
   cost(0, firm) = cost(scen0, firm)
NEXT firm
RETURN
'***** END OF InitProb12 SUBMODULE *****


INITIALIZE3:
'Submodule to initialize splitting Microsoft into three firms.
GOSUB INITPROB13:
OPEN prob3$ FOR OUTPUT AS #2
GOSUB PRINTPROB0:
RETURN
'***** END OF Initialize3 SUBMODULE *****


INITPROB13:
'This submodule sets the prob1 variables to zero,
'   and then sets initial values for non-zero prob1.
FOR scen1 = 0 TO 35
FOR firm1 = 0 TO 3
   prob1(scen1, firm1) = 0
NEXT firm1
NEXT scen1
'This section sets initial scenario to
'   Microsoft is split into three equal-sized firms,
'   Scenario 7, Cost levels 3(MS-1), 4(MS-2), 4(MS-3).
scen0 = 7
prob1(0, 1) = 1# / 3#
prob1(0, 2) = 1# / 3#
prob1(0, 3) = 1# / 3#
FOR firm = 1 TO 3
```

```
   cost(0, firm) = cost(scen0, firm)
NEXT firm
RETURN
'***** END OF InitProb13 SUBMODULE *****


SCENREAD:
'This submodule reads in the scenario costs list.
OPEN cost$ FOR INPUT AS #1
LINE INPUT #1, dummy$
'cost(s,f) = short-run marginal cost of firm f in scenario s.
'finprob(s) = final probability assumed for scenario s.
'wgt(scen) = number of permutations of scenario s.
FOR scen = 1 TO 35
   INPUT #1, scen2, L1, L2, L3, L4, L5, wgt, finprob(scen)
     IF scen <> scen2 THEN PRINT "Scenario mismatch", scen, scen2
   INPUT #1, c1, c2, c3
   FOR firm = 1 TO 3
     INPUT #1, ctemp
     cost(scen, firm) = ctemp / 60#
   NEXT firm
NEXT scen
CLOSE #1
RETURN
'***** END OF ScenRead SUBMODULE *****


ORDERREAD:
'This Submodule reads in the ordering list,
'   which is a list of 6 permutations by which firms 1-3
'   may become firms 1-3 in the same or a different order.
OPEN order$ FOR INPUT AS #1
LINE INPUT #1, dummy$
'ordnum = number of ordering.
'order(o,f) = ordering number o for firm f,
'   the firm number which firm f becomes in ordering o.
DIM order(6, 3)
FOR ordnum = 1 TO 6
   INPUT #1, ordnum2
     IF ordnum <> ordnum2 THEN PRINT "Order Number mismatch",
ordnum, ordnum2
   FOR firm = 1 TO 3
     INPUT #1, order(ordnum, firm)
   NEXT firm
```

```
NEXT ordnum
CLOSE #1
RETURN
'***** END OF OrderRead SUBMODULE *****


TRANSIT0:
'This submodule controls the initial transitions.
'   Transitions are from scenario zero (scen1)
'   to the other possible scenarios (scen2).
GOSUB INITPROB2: 'Initialize prob2 variables.
PRINT speed; iter; "*";
scen1 = 0
      PRINT ".";
   iprob0 = prob1(scen1, 0)
   iprob1 = prob1(scen1, 1)
   iprob2 = prob1(scen1, 2)
   iprob3 = prob1(scen1, 3)
FOR scen2 = 1 TO 35
   GOSUB PVADD:
NEXT scen2
   GOSUB PVADJUST:
GOSUB MSEXITS:
PRINT
RETURN
'***** END OF Transit0 SUBMODULE *****


TRANSIT1:
'This submodule controls the subsequent transitions.
'   Transitions are from any scenario (scen1)
'   to any same or different scenario (scen2).
GOSUB INITPROB2: 'Initialize prob2 variables.
PRINT speed; iter; "*";
FOR scen1 = 1 TO 35
      PRINT ".";
   iprob0 = prob1(scen1, 0)
   iprob1 = prob1(scen1, 1)
   iprob2 = prob1(scen1, 2)
   iprob3 = prob1(scen1, 3)
IF iprob0 + iprob1 + iprob2 + iprob3 = 0 THEN 10
FOR scen2 = 1 TO 35
   GOSUB PVADD:
NEXT scen2
```

```
   GOSUB PVADJUST:
10 NEXT scen1
GOSUB MSEXITS:
PRINT
RETURN
'***** END OF Transit1 SUBMODULE *****


INITPROB2:
'This submodule of TRANSIT sets the prob2 variables to zero.
FOR scen2 = 0 TO 35
FOR firm2 = 0 TO 3
   prob2(scen2, firm2) = 0
NEXT firm2
NEXT scen2
RETURN
'***** END OF InitProb2 SUBMODULE *****


PVADD:
'This submodule adds up point values (pv) for transition
'   from a single scenario (scen1)
'   to a single scenario (scen2).
'pvtot0(s,f) = point value for probability of transition
'   from scenario with Microsoft=firm 0 (zero market share),
'   to scenario s and to Microsoft=firm f.
'pvtot1(s,f) = same, but from Microsoft=firm 1.
'pvtot2(s,f) = same, but from Microsoft=firm 2.
'pvtot3(s,f) = same, but from Microsoft=firm 3.
FOR firm1 = 1 TO 3
  FOR firm2 = 1 TO 3
    diff(firm1, firm2) = ABS(cost(scen1, firm1) - cost(scen2,
firm2))
  NEXT firm2
NEXT firm1

FOR firm2 = 0 TO 3
  pvtot0(scen2, firm2) = 0
  pvtot1(scen2, firm2) = 0
  pvtot2(scen2, firm2) = 0
  pvtot3(scen2, firm2) = 0
NEXT firm2
sprob = iwgt(scen2)
sprob3 = sprob / 6#
```

```
GOSUB PVADD3:
RETURN
'***** END OF PVadd SUBMODULE *****


PVADD3:
'This submodule of PVADD adds up point values for transition
'   from a single scenario (scen1)
'   to a single scenario (scen2),
'   where scen1 and scen2 both have 3 firms.
FOR o = 1 TO 6
  o1 = order(o, 1)
  o2 = order(o, 2)
  o3 = order(o, 3)
  pv = 1
  pvtemp = pvmax - diff(1, o1)
  IF pvtemp < 0 THEN pvtemp = 0
  pv = pv * pvtemp
  pvtemp = pvmax - diff(2, o2)
  IF pvtemp < 0 THEN pvtemp = 0
  pv = pv * pvtemp
  pvtemp = pvmax - diff(3, o3)
  IF pvtemp < 0 THEN pvtemp = 0
  pv = pv * pvtemp
  pvtot0(scen2, 0) = pvtot0(scen2, 0) + pv * iprob0 * sprob3
  pvtot1(scen2, o1) = pvtot1(scen2, o1) + pv * iprob1 * sprob3
  pvtot2(scen2, o2) = pvtot2(scen2, o2) + pv * iprob2 * sprob3
  pvtot3(scen2, o3) = pvtot3(scen2, o3) + pv * iprob3 * sprob3
NEXT o
RETURN
'***** END OF PVadd3 SUBMODULE *****


PVADJUST:
'This module adjusts computed point values (pv)
'   to reflect true probability measures (prob2).
pvtotal0 = 0
pvtotal1 = 0
pvtotal2 = 0
pvtotal3 = 0
FOR scen = 1 TO 35
  FOR firm2 = 0 TO 3
    pvtotal0 = pvtotal0 + pvtot0(scen, firm2)
    pvtotal1 = pvtotal1 + pvtot1(scen, firm2)
```

```
    pvtotal2 = pvtotal2 + pvtot2(scen, firm2)
    pvtotal3 = pvtotal3 + pvtot3(scen, firm2)
  NEXT firm2
NEXT scen

ratio0 = 0
ratio1 = 0
ratio2 = 0
ratio3 = 0
20 IF pvtotal0 = 0 THEN 21
ratio0 = prob1(scen1, 0) / pvtotal0
21 IF pvtotal1 = 0 THEN 22
ratio1 = prob1(scen1, 1) / pvtotal1
22 IF pvtotal2 = 0 THEN 23
ratio2 = prob1(scen1, 2) / pvtotal2
23 IF pvtotal3 = 0 THEN 24
ratio3 = prob1(scen1, 3) / pvtotal3
24 REM

FOR scen = 1 TO 35
  FOR firm2 = 0 TO 3
    pvtemp = pvtot0(scen, firm2) * ratio0
    pvtemp = pvtemp + pvtot1(scen, firm2) * ratio1
    pvtemp = pvtemp + pvtot2(scen, firm2) * ratio2
    pvtemp = pvtemp + pvtot3(scen, firm2) * ratio3
    prob2(scen, firm2) = prob2(scen, firm2) + pvtemp
  NEXT firm2
NEXT scen

RETURN
'***** END OF PVadjust SUBMODULE *****


MSEXITS:
'This submodule determines which prob2(s,f) and cost(s,f)
'   numbers imply zero market share for Microsoft.
'   Where this occurs for f>0 (MS still in market),
'   the probability values are transferred
'   to f=0 (Microsoft not in market).
'   The criterion for exit is that the firm in question
'   has very high short-run costs.
FOR scen = 1 TO 35
FOR firm = 1 TO 3
   IF cost(scen, firm) > 4.999 THEN
```

```
      prob2(scen, 0) = prob2(scen, 0) + prob2(scen, firm)
      prob2(scen, firm) = 0
      END IF
NEXT firm
NEXT scen
RETURN
'***** END OF MSexits SUBMODULE *****


PRINTPROBT:
'This submodule prints the last iteration (presumed convergence)
'   computed for the the transition weights for each scenario.
PRINT #2, " Iter "; "Scen ";
PRINT #2, "Init-weight(0) "; "Prob1(target)   "; "Prob2(result)
"; "Init-weight(1) "
FOR scen = 1 TO 35
   PRINT #2, USING "#####"; iter; scen;
   PRINT #2, USING "##.###########"; iwgt0(scen); prob1(scen,
0); prob2(scen, 0); iwgt(scen)
NEXT scen
RETURN
'***** END OF PrintProbT SUBMODULE *****


PRINTPROB0:
'This submodule prints the probabilities for scenario zero.
PRINT #2, " Iter "; "Scen ";
PRINT #2, "Prob(firm0)     "; "Prob(firm1)     "; "Prob(firm2)
";
PRINT #2, "Prob(firm3)     "
scen = 0
   PRINT #2, USING "#####"; iter; scen0;
   FOR firm = 0 TO 3
      PRINT #2, USING "##.###########"; prob1(scen, firm);
   NEXT firm
   PRINT #2,
RETURN
'***** END OF PrintProb0 SUBMODULE *****


PRINTPROB:
'This submodule prints the probabilities for each subsequent
'   scenario and MS firm number.
FOR scen = 1 TO 35
```

```
   PRINT #2, USING "#####"; iter; scen;
   FOR firm = 0 TO 3
      PRINT #2, USING "##.###########"; prob2(scen, firm);
   NEXT firm
   PRINT #2,
NEXT scen
RETURN
'***** END OF PrintProb SUBMODULE *****


TRANSFERPROB:
'This submodule transfers the prob2 values to prob1,
'   so that the next transition iteration can proceed.
FOR scen = 0 TO 35
FOR firm = 0 TO 3
   prob1(scen, firm) = prob2(scen, firm)
NEXT firm
NEXT scen
RETURN
'***** END OF TransferProb SUBMODULE *****

'**********END OF Program "MS2ProbA.bas".**********
```

Attachment N.

```
'BASIC Program "MS3ProbR.bas".

'Program Number 3 in a series of six programs
'designed to simulate alternative antitrust
'remedies for the Microsoft software industry.

'Copyright, January 23, 2002, Carl Lundgren.

'   This program, "MS3ProbR.bas", computes the probabilites
'associated with each scenario, as the industry transitions
'from a particular starting point, and gradually converges
'towards a long-run stochastic equilibrium.
'This program assigns probabilities for equilibria consisting
'of two Relative Profit Maximizing ("RPM") firms, along with
'such Absolute Profit Maximizing "APM" firms as may be involved
'in the transitions.  The equilibria automatically convert to
'APM equilibria if one or both RPM firms exits the industry.
'The program uses three different speeds (speed=1,2,3) for the
transition.
'Probability files are outputted for the one starting point
'(msfirms=2), and each transition speed (speed=1,2,3).

'   This program is similar to "MS2ProbA.bas",
'since it computes probabilites associated with each scenario,
'for a total of 11 transition periods.
'This program differs from "MS2ProbA.bas",
'because it assigns probabilities for equilibria consisting
'of both RPM and APM firms, rather than APM firms only.
'This program is more complex than "MS2ProbA.bas"
'because it must simultaneously track the rankings
'of two Microsoft-successor firms simultaneously.

'   This program calculates transition probabilities
'where Microsoft starts as two firms, and simultaneously
'tracks the outcomes and rankings for both firms.
'   The parameters controlling the transition speed
'(pvmax in submodule InitProb10) are supplied by the user.
'The program reads in 35 possible cost structures
'for the industry, each with 3 firms.
'The program assigns probabilities for each scenario,
'and also tracks whether Microsoft #1 is ranked as
'firm 1, 2, or 3, or is firm 0 (with zero market share).
```

```
'Likewise, the program tracks whether Microsoft #2 is
'ranked as firm 1, 2, or 3, or is firm 0.

DEFDBL A-Z
DIM pvtot(35, 15), prob1r(35, 15), prob2r(35, 15)
DIM diff(3, 3), iwgt(35), cost(35, 3)


'CONTROL MODULE
CLS
timex = TIMER
'This section calls the main module 3 times.
'This control module chooses speed for cost shifts:
   'speed = 1 'Slow speed for cost shifts.
   'speed = 2 'Moderate speed for cost shifts.
   'speed = 3 'High speed for cost shifts.
FOR speed = 1 TO 3
   GOSUB MAINMODULE:
NEXT speed
PRINT TIMER - timex
END


MAINMODULE:
IF speed = 1 THEN pvmax = 1.5 'Slow speed for cost shifts.
IF speed = 2 THEN pvmax = 2.5 'Moderate speed for cost shifts.
IF speed = 3 THEN pvmax = 4.5 'High speed for cost shifts.
GOSUB FILENAMES: 'Assign file names to input/output files.
GOSUB SCENREAD: 'Read in scenario costs.
GOSUB ORDERREAD: 'Read in ordering list.
GOSUB ORDERRPMREAD: 'Read in orderRPM list.
GOSUB READIWGT: 'Read in values for transition weights.

PRINT "Computing transitions from MS=2 RPM firms:"
msfirms = 2
iter = 0
GOSUB INITIALIZER: 'Microsoft split into 2 RPM firms.
iter = 1
GOSUB TRANSIT0R:
GOSUB PRINTPROBR:
FOR iter = 2 TO 10
   GOSUB TRANSFERPROBR:
   GOSUB TRANSIT1R:
   GOSUB PRINTPROBR:
```

```
NEXT iter
CLOSE #2, #3

CLOSE
RETURN
'***** END OF MAIN MODULE *****


FILENAMES:
cost$ = "c:\basic\ms_sim\costlist.txt" 'Input scenario costs
order$ = "c:\basic\ms_sim\ordering.txt" 'Input firm re-orderings
orderrpm$ = "c:\basic\ms_sim\orderrpm.txt" 'Input RPM firm-pair
re-orderings
prob0$ = "c:\basic\ms_sim\out\prob00.txt" 'Input 2 RPM firms I-
weight probs
probr$ = "c:\basic\ms_sim\out\probr0.txt" 'Output 2 RPM firms
transition probs
IF speed = 1 THEN sp$ = "1"
IF speed = 2 THEN sp$ = "2"
IF speed = 3 THEN sp$ = "3"
replace$ = sp$
MID$(prob0$, 26, 1) = replace$
MID$(probr$, 26, 1) = replace$
RETURN
'***** END OF FileNames SUBMODULE *****


READIWGT:
'This submodule reads in the transition weights (iwgt)
'   previously computed by the "MS2ProbA.bas" program.
OPEN prob0$ FOR INPUT AS #1
LINE INPUT #1, temp$
FOR scen = 1 TO 35
   INPUT #1, iter2, scen2, iwgt0, prob1scen, prob2scen,
iwgt(scen)
   IF scen2 <> scen THEN PRINT "Scenario mismatch:"; scen; scen2
NEXT scen
CLOSE #1
RETURN
'***** END OF ReadIwgt SUBMODULE *****


INITIALIZER:
'Submodule to initialize Microsoft split into 2 RPM firms.
```

```
GOSUB INITPROB1R:
OPEN probr$ FOR OUTPUT AS #2
GOSUB PRINTPROB0R:
RETURN
'***** END OF InitializeR SUBMODULE *****


INITPROB1R:
'This submodule sets the prob1r variables to zero,
'   and then sets initial values for non-zero prob1.
FOR scen1 = 0 TO 35
FOR pair1 = 0 TO 15
   prob1r(scen1, pair1) = 0
NEXT pair1
NEXT scen1
'This section sets initial scenario to
'   Microsoft is split into two RPM firms,
'   Scenario 6, Cost levels 3(MS-1), 4(MS-2), 5(comp).
scen0 = 6
firm1 = 1
firm2 = 2
pair = firm1 * 4 + firm2
prob1r(0, pair) = 1
FOR firm = 1 TO 3
   cost(0, firm) = cost(scen0, firm)
NEXT firm
RETURN
'***** END OF InitProb12 SUBMODULE *****


SCENREAD:
'This submodule reads in the scenario costs list.
OPEN cost$ FOR INPUT AS #1
LINE INPUT #1, dummy$
'cost(s,f) = marginal cost of firm f in scenario s.
'finprob = final probability assumed for scenario s.
'wgt = number of permutations of scenario s.
FOR scen = 1 TO 35
   INPUT #1, scen2, L1, L2, L3, L4, L5, wgt, finprob
     IF scen <> scen2 THEN PRINT "Scenario mismatch", scen, scen2
   INPUT #1, c1, c2, c3
   FOR firm = 1 TO 3
     INPUT #1, ctemp
     cost(scen, firm) = ctemp / 60#
```

```
   NEXT firm
NEXT scen
CLOSE #1
RETURN
'***** END OF SCENREAD SUBMODULE *****


ORDERREAD:
'This Submodule reads in the ordering list,
'   which is a list of 6 permutations by which firms 1-3
'   may become firms 1-3 in the same or a different order.
OPEN order$ FOR INPUT AS #1
LINE INPUT #1, dummy$
'ordnum = number of ordering.
'order(o,f) = ordering number o for firm f,
'   the firm number which firm f becomes in ordering o.
DIM order(6, 3)
FOR ordnum = 1 TO 6
   INPUT #1, ordnum2
     IF ordnum <> ordnum2 THEN PRINT "Order Number mismatch",
ordnum, ordnum2
   FOR firm = 1 TO 3
     INPUT #1, order(ordnum, firm)
   NEXT firm
NEXT ordnum
CLOSE #1
RETURN
'***** END OF OrderRead SUBMODULE *****


ORDERRPMREAD:
'This Submodule reads in the orderRPM list,
'   which is a list of 6 permutations by which firms 1-3
'   may become firms 1-3 in the same or a different order.
'The orderRPM list simultaneously tracks the cost rankings
'   of two RPM firms.
OPEN orderrpm$ FOR INPUT AS #1
LINE INPUT #1, dummy$
'ordnum = number of ordering.
'orderrpm(o,f) = ordering number o for pair of firms p,
'   the firm-pair number to which the firm-pair p
'   becomes in ordering o.
'p is firm-pair where p=4*firm1+firm2.
'Firm1 and firm2 take on values (0, 1, 2, 3).
```

```
DIM orderrpm(6, 15)
FOR ordnum = 1 TO 6
   INPUT #1, ordnum2
     IF ordnum <> ordnum2 THEN PRINT "Order Number mismatch",
ordnum, ordnum2
   FOR pair = 0 TO 15
     INPUT #1, orderrpm(ordnum, pair)
   NEXT pair
NEXT ordnum
CLOSE #1
RETURN
'***** END OF OrderRPMread SUBMODULE *****


TRANSIT0R:
'This submodule controls the initial transitions.
'   Transitions are from scenario zero (scen1)
'   to the other possible scenarios (scen2).
GOSUB INITPROB2R: 'Initialize prob2 variables.
PRINT speed; iter; "*";
scen1 = 0
      PRINT ".";
FOR pair1 = 0 TO 15
   iprob = prob1r(scen1, pair1)
   IF iprob = 0 THEN 10
   FOR scen2 = 1 TO 35
      GOSUB PVADDR:
   NEXT scen2
   GOSUB PVADJUSTR:
10 NEXT pair1
GOSUB MSEXITSR:
PRINT
RETURN
'***** END OF Transit0R SUBMODULE *****


TRANSIT1R:
'This submodule controls the subsequent transitions.
'   Transitions are from any scenario (scen1)
'   to any same or different scenario (scen2).
GOSUB INITPROB2R: 'Initialize prob2 variables.
PRINT speed; iter; "*";
FOR scen1 = 1 TO 35
      PRINT ".";
```

```
FOR pair1 = 0 TO 15
   iprob = prob1r(scen1, pair1)
   IF iprob = 0 THEN 11
   FOR scen2 = 1 TO 35
      GOSUB PVADDR:
   NEXT scen2
   GOSUB PVADJUSTR:
11 NEXT pair1
NEXT scen1
GOSUB MSEXITSR:
PRINT
RETURN
'***** END OF Transit1R SUBMODULE *****


INITPROB2R:
'This submodule of TRANSIT sets the prob2r variables to zero.
FOR scen2 = 0 TO 35
FOR pair2 = 0 TO 15
   prob2r(scen2, pair2) = 0
NEXT pair2
NEXT scen2
RETURN
'***** END OF InitProb2R SUBMODULE *****


PVADDR:
'This submodule initializes the variables in preparation
'   for submodule PVADD3R,
'   which adds up point values (pv) for transition
'   from a single scenario (scen1) and firm pair (pair1)
'   to a single scenario (scen2) and multiple pairs (pair2).
'pvtot(s,p) = point value for probability of transition
'   from current scenario and current MS firm pair
'   to scenario s and to MS firm pair p.
FOR firm1 = 1 TO 3
  FOR firm2 = 1 TO 3
    diff(firm1, firm2) = ABS(cost(scen1, firm1) - cost(scen2,
firm2))
  NEXT firm2
NEXT firm1

FOR pair2 = 0 TO 15
  pvtot(scen2, pair2) = 0
```

```
NEXT pair2
sprob = iwgt(scen2)
sprob3 = sprob / 6#
GOSUB PVADD3R:
RETURN
'***** END OF PVaddR SUBMODULE *****


PVADD3R:
'This submodule of PVADDR adds up point values for transition
'   from a single scenario (scen1) and firm pair (pair1)
'   to a single scenario (scen2) and multiple firm pairs (pair2).
FOR o = 1 TO 6
  o1 = order(o, 1)
  o2 = order(o, 2)
  o3 = order(o, 3)
  pv = 1
  pvtemp = pvmax - diff(1, o1)
  IF pvtemp < 0 THEN pvtemp = 0
  pv = pv * pvtemp
  pvtemp = pvmax - diff(2, o2)
  IF pvtemp < 0 THEN pvtemp = 0
  pv = pv * pvtemp
  pvtemp = pvmax - diff(3, o3)
  IF pvtemp < 0 THEN pvtemp = 0
  pv = pv * pvtemp

  orpm = orderrpm(o, pair1)
  pvtot(scen2, orpm) = pvtot(scen2, orpm) + pv * iprob * sprob3
NEXT o
RETURN
'***** END OF PVadd3R SUBMODULE *****


PVADJUSTR:
'This module adjusts computed point values (pv)
'   to reflect true probability measures (prob2).
pvtotal = 0
FOR scen = 1 TO 35
  FOR pair2 = 0 TO 15
    pvtotal = pvtotal + pvtot(scen, pair2)
  NEXT pair2
NEXT scen
```

```
ratio = 0
20 IF pvtotal = 0 THEN 21
ratio = prob1r(scen1, pair1) / pvtotal
21 REM

FOR scen = 1 TO 35
  FOR pair2 = 0 TO 15
    probtemp = pvtot(scen, pair2) * ratio
    prob2r(scen, pair2) = prob2r(scen, pair2) + probtemp
  NEXT pair2
NEXT scen

RETURN
'***** END OF PVadjustR SUBMODULE *****


MSEXITSR:
'This submodule determines which prob2r(s,f) and cost(s,f)
'   numbers imply exiting the industry for Microsoft
'   or a Microsoft successor.
'   Where this occurs for firm1>0 (MS #1 still in market)
'   or for firm2>0 (MS #2 still in market),
'   the probability values are transferred respectively
'   to firm1=0 (Microsoft #1 not in market) or
'   to firm2=0 (Microsoft #2 not in market).
'   The criterion for exit is that the firm in question
'   has very high short-run costs.
FOR scen = 1 TO 35
FOR firm1 = 1 TO 3
FOR firm2 = 0 TO 3
   pair = firm1 * 4 + firm2
   IF cost(scen, firm1) > 4.999 THEN
      pair0 = firm2'firm1=0
      prob2r(scen, pair0) = prob2r(scen, pair0) + prob2r(scen,
pair)
      prob2r(scen, pair) = 0
      END IF
NEXT firm2
NEXT firm1
FOR firm1 = 0 TO 3
FOR firm2 = 1 TO 3
   pair = firm1 * 4 + firm2
   IF cost(scen, firm2) > 4.999 THEN
      pair0 = firm1 * 4 'firm2=0
```

```
      prob2r(scen, pair0) = prob2r(scen, pair0) + prob2r(scen,
pair)
      prob2r(scen, pair) = 0
      END IF
NEXT firm2
NEXT firm1
NEXT scen
RETURN
'***** END OF MSexitsR SUBMODULE *****


PRINTPROB0R:
'This submodule prints the probabilities
'   for each firm-pair number for scenario zero.
PRINT #2, " Iter "; "Scen "; "Firm ";
PRINT #2, "Prob(firm0)    "; "Prob(firm1)    "; "Prob(firm2)
";
PRINT #2, "Prob(firm3)    "
scen = 0
FOR firm1 = 0 TO 3
   PRINT #2, USING "#####"; iter; scen0; firm1;
   FOR firm2 = 0 TO 3
      pair = 4 * firm1 + firm2
      PRINT #2, USING "##.###########"; prob1r(scen, pair);
   NEXT firm2
   PRINT #2,
NEXT firm1
RETURN
'***** END OF PrintProb0 SUBMODULE *****


PRINTPROBR:
'This submodule prints the probabilities
'   for each MS firm-pair number
'   for each subsequent scenario.
FOR scen = 1 TO 35
FOR firm1 = 0 TO 3
   PRINT #2, USING "#####"; iter; scen; firm1;
   FOR firm2 = 0 TO 3
      pair = 4 * firm1 + firm2
      PRINT #2, USING "##.###########"; prob2r(scen, pair);
   NEXT firm2
   PRINT #2,
NEXT firm1
```

```
NEXT scen
RETURN
'***** END OF PrintProb SUBMODULE *****


TRANSFERPROBR:
'This submodule transfers the prob2 values to prob1,
'   so that the next transition iteration can proceed.
FOR scen = 0 TO 35
FOR pair = 0 TO 15
   prob1r(scen, pair) = prob2r(scen, pair)
NEXT pair
NEXT scen
RETURN
'***** END OF TransferProbR SUBMODULE *****

'**********END OF Program "MS3ProbR.bas".**********
```

Attachment O.

```
'BASIC Program "MS4TranA.bas".

'Program Number 4 in a series of six programs
'designed to simulate alternative antitrust
'remedies for the Microsoft software industry.

'Copyright, January 23, 2002, Carl Lundgren.

'   This program, "MS4TranA.bas", uses the probability data
'computed by "MS2ProbA.bas" to compute Consumer Surplus and
'Profits for both Microsoft and Microsoft's competitors.
'In transition period zero (iter=0), Microsoft (and its
'successor firms after divestiture) are assumed to have no
competitors.
'In subsequent transition periods (iter=1 to 10),
'Microsoft has (potentially) one or more competitors.
'This program only calculates Absolute Profit Maximizing ("APM")
equilibria.
'   The program uses the computed probabilities for each
'scenario that was previously outputted by the
'"MS2ProbA.bas" program as various "PROB....txt" files.
'   This program outputs as "TRAN....txt" files the
'computed transition factors for several alternative timepaths
'for the software industry, under several alternative
assumptions.
'These transition factors are computed as a fraction
'of the revenues which Microsoft would earn if it remained
'a monopoly.  The assumptions for the transitions are:
'   Tran1) Strong conduct remedy & Lawful Path:
'Microsoft is not broken up, but competitive conditions
'start in transition period zero.  A companion
'program, "MS6Summ.bas", uses the transition factors
'to compute the lawful path (starting in 1995) and
'a conduct remedy (starting in 2002).
'   Tran2-Tran3) APM Structural remedies:
'Microsoft is broken up into two or three competing APM firms,
'beginning in transition period zero.  The companion program
'uses these transition factors to compute the effects of
'structural remedies starting in 2005.
'   The 225 outputted transition (TRAN....txt) files are computed
'for three speeds of transition (speed=1,2,3),
```

```
'five cost ratios for short-run cost (cratio=1,2,3,4,5),
'five assumptions concerning the portion of long-run costs
'allocated to fixed costs (port=0,1,2,3,4),
'and three starting points (msfirms=1,2,3).

DEFDBL A-Z
DIM probl(35, 3), herf(35), mshare(35, 3), pnum(35)
DIM quant(35, 3), cost(35, 3), pv(35, 3), price(35)
DIM pims(35, 3), picomp(35, 3)


'CONTROL MODULE
CLS
timex = TIMER
GOSUB SCENREAD:
'This section calls the main module 225 times.
'This control module chooses market tendency:
   'cratio=1 'Ratio for low/high short-run cost is 0.2500
(1/4.0).
   'cratio=2 'Ratio for low/high short-run cost is 0.3333
(1/3.0).
   'cratio=3 'Ratio for low/high short-run cost is 0.4000
(1/2.5).
   'cratio=4 'Ratio for low/high short-run cost is 0.5000
(1/2.0).
   'cratio=5 'Ratio for low/high short-run cost is 0.6667
(1/1.5).
'This control module chooses speed for market share shifts:
   'speed = 1 'Slow speed for market share shifts.
   'speed = 2 'Moderate speed for market share shifts.
   'speed = 3 'High speed for market share shifts.
'This control module chooses # of msfirms at iteration zero.
   'msfirms = 1 'Microsoft starts as a monopoly.
   'msfirms = 2 'Microsoft split into 2 APM firms.
   'msfirms = 3 'Microsoft split into 3 APM firms.
'This control module chooses proportion of long-run cost
'   which is assumed to be a fixed cost.
   'port = 0 'Fixed cost is 0% of long-run cost.
   'port = 1 'Fixed cost is 25% of long-run cost.
   'port = 2 'Fixed cost is 50% of long-run cost.
   'port = 3 'Fixed cost is 75% of long-run cost.
   'port = 4 'Fixed cost is 100% of long-run cost.
```

```
FOR cratio = 1 TO 5
   FOR speed = 1 TO 3
     FOR port = 0 TO 4
       FOR msfirms = 1 TO 3
         GOSUB MAINMODULE:
       NEXT msfirms
     NEXT port
   NEXT speed
NEXT cratio
PRINT TIMER - timex
END


MAINMODULE:
GOSUB FILENAMES: 'Assign file names to input/output files.
GOSUB INITIALIZE:
FOR iter = 1 TO 10
   GOSUB PROBREAD:
   GOSUB PRINTTRAN:
NEXT iter
CLOSE
RETURN
'***** END OF MAIN MODULE *****


FILENAMES:
prob$ = "c:\basic\ms_sim\out\prob00.txt" 'Input transition
probabilities
tran$ = "c:\basic\ms_sim\out\tran0000.txt" 'Output transition
factors
IF cratio = 1 THEN crt$ = "1"
IF cratio = 2 THEN crt$ = "2"
IF cratio = 3 THEN crt$ = "3"
IF cratio = 4 THEN crt$ = "4"
IF cratio = 5 THEN crt$ = "5"
IF speed = 1 THEN sp$ = "1"
IF speed = 2 THEN sp$ = "2"
IF speed = 3 THEN sp$ = "3"
IF msfirms = 1 THEN msf$ = "1"
IF msfirms = 2 THEN msf$ = "2"
IF msfirms = 3 THEN msf$ = "3"
IF port = 0 THEN prt$ = "0"
IF port = 1 THEN prt$ = "1"
```

```
IF port = 2 THEN prt$ = "2"
IF port = 3 THEN prt$ = "3"
IF port = 4 THEN prt$ = "4"
replacep$ = msf$ + sp$
replacet$ = msf$ + crt$ + sp$ + prt$
MID$(prob$, 25, 2) = replacep$
MID$(tran$, 25, 4) = replacet$
PRINT replacet$; " ";
RETURN
'***** END OF FileNames SUBMODULE *****


INITIALIZE:
'Submodule to perform various initialization tasks.
OPEN prob$ FOR INPUT AS #2
OPEN tran$ FOR OUTPUT AS #3
GOSUB ZEROPROB:
GOSUB PROBREAD0:
GOSUB SCENREAD: 'Read scenario list.
GOSUB COSTCOMPUTE: 'Compute costs.
GOSUB PQZERO: 'Iteration 0 prices, quantities, profits & Consumer
Surplus.
GOSUB PQCOMPUTE: 'Compute prices, quantities, profits & Consumer
Surplus.
GOSUB HHI: 'Compute HHI and market shares.
GOSUB PROFITS: 'Assign profits to MS and competitors.
GOSUB PRINTTRAN0: 'Print transition files.
RETURN
'***** END OF Initialize SUBMODULE *****


ZEROPROB:
'This submodule sets the probl(0, .) and mshare(0, .)
'   variable values to zero.
FOR firml = 0 TO 3
   probl(0, firml) = 0
NEXT firml
RETURN
'***** END OF ZEROPROB SUBMODULE *****


SCENREAD:
'This submodule reads in the scenario costs list.
```

```
cost$ = "c:\basic\ms_sim\costlist.txt" 'Input scenario costs
OPEN cost$ FOR INPUT AS #1
LINE INPUT #1, dummy$
'cost(s,f) = short-run marginal cost of firm f in scenario s.
'finprob(s) = final probability assumed for scenario s.
'wgt(scen) = number of permutations of scenario s.
FOR scen = 1 TO 35
   INPUT #1, scen2, L1, L2, L3, L4, L5, wgt, finprob
     IF scen <> scen2 THEN PRINT "Scenario mismatch", scen, scen2
   INPUT #1, c1, c2, c3
   FOR firm = 1 TO 3
     INPUT #1, pv(scen, firm)
   NEXT firm
NEXT scen
CLOSE #1
RETURN
'***** END OF SCENREAD SUBMODULE *****


COSTCOMPUTE:
'Submodule to compute short-run costs, long-run costs,
'   and assumed elasticity of demand.

'This section computes parameters for long-run costs
'   under the assumption that each firm has
'   the same long-run cost function.
'Assume that one portion of Microsoft's
'   long-run cost (LRC) is a long-run fixed cost (FC),
'   while the other portion is a long-run variable cost (VC),
'   which is proportional to output.
lrc = .1855 'computed as MS long-run cost divided by MS monopoly
revenue.
IF port = 0 THEN portion = 0!
IF port = 1 THEN portion = .25
IF port = 2 THEN portion = .5
IF port = 3 THEN portion = .75
IF port = 4 THEN portion = 1!
fc = lrc * portion
vc = lrc * (1 - portion)

'This section computes elasticity of demand (Elas) at
'   monopoly profit maximum, as a function of marginal cost,
'   which is composed of short-run marginal cost (SRC)
```

```
'   plus long-run variable cost (VC).
src = .4101 'computed as MS short-run cost divided by MS monopoly
revenue.
mc = src + vc
elas = 1 / (mc - 1)
elasminus = elas - 1
elasplus = elas + 1
A = elasminus / elas 'Intercept of linear demand curve with price
axis.
b = -1 / elas 'Slope of linear demand curve.
cbase = src 'Base level of short-run marginal cost (cost level
2).
logcbase = LOG(cbase / (A - vc - cbase)) 'Cbase converted to log-
ratios.

'This section computes short-run costs and marginal costs
'   for a given cost spread.
IF cratio = 1 THEN cspread = .950980935#
IF cratio = 2 THEN cspread = .748669813#
IF cratio = 3 THEN cspread = .622288438#
IF cratio = 4 THEN cspread = .469161475#
IF cratio = 5 THEN cspread = .273626703#
FOR scen = 1 TO 35
   FOR firm = 1 TO 3
      pvtemp = (pv(scen, firm) - 180) / 120
      logpv = logcbase + pvtemp * cspread
      pvratio = EXP(logpv)
      cost(scen, firm) = vc + (A - vc) * pvratio / (1 + pvratio)
   NEXT firm
NEXT scen
RETURN
'***** END OF CostCompute SUBMODULE *****


PQZERO:
'Submodule to compute prices, quantities, profits,
'   and consumer surplus for selected scenarios,
'   for iteration zero, where 1, 2, or 3 Microsoft APM firms
'   are assumed initially to have no competitors.
'Pi(s,f) is long-run profit for firm f within scenario s.
'CS(s) is Consumer Surplus parameter for scenario s.
DIM cs(35), pi(35, 3)
   num = msfirms
```

```
    costsum = 0
    FOR firm = 1 TO num
       cost(0, firm) = cost(scen0, firm)
       costsum = costsum + cost(0, firm)
    NEXT firm
    price = (A + costsum) / (num + 1)
    qtot = 0
    FOR firm = 1 TO num
       qtemp = (price - cost(0, firm)) / b
       qtot = qtot + qtemp
       quant(0, firm) = qtemp
       pitemp = (price - cost(0, firm)) * qtemp
       pitemp = pitemp - fc
       pi(0, firm) = pitemp
    NEXT firm
    FOR firm = num + 1 TO 3
       quant(0, firm) = 0
       pi(0, firm) = 0
    NEXT firm
    cs(0) = qtot * (A - price) / 2
    pdummy = 1 'Is last firm producing?
    IF quant(0, num) < 0 THEN pdummy = 0
    IF pi(0, num) < 0 THEN pdummy = 0
    IF pdummy = 0 THEN
       scen = 0
       IF msfirms = 3 THEN GOSUB PQSUB2:
       IF msfirms = 2 THEN GOSUB PQSUB1:
       IF msfirms = 1 THEN GOSUB PQSUB0:
       END IF
    price(0) = price
    pnum(0) = num
RETURN
'***** END OF PQzero SUBMODULE *****


PQCOMPUTE:
'Submodule to compute prices, quantities, profits,
'   and consumer surplus for each scenario.
'Pi(s,f) is long-run profit for firm f within scenario s.
'CS(s) is Consumer Surplus parameter for scenario s.
FOR scen = 1 TO 35
   num = 3
   costsum = 0
```

```
    FOR firm = 1 TO num
       costsum = costsum + cost(scen, firm)
    NEXT firm
    price = (A + costsum) / (num + 1)
    qtot = 0
    FOR firm = 1 TO num
       qtemp = (price - cost(scen, firm)) / b
       qtot = qtot + qtemp
       quant(scen, firm) = qtemp
       pitemp = (price - cost(scen, firm)) * qtemp
       pitemp = pitemp - fc
       pi(scen, firm) = pitemp
    NEXT firm
    cs(scen) = qtot * (A - price) / 2
    pdummy = 1 'Is last firm producing?
    IF quant(scen, num) < 0 THEN pdummy = 0
    IF pi(scen, num) < 0 THEN pdummy = 0
    IF pdummy = 0 THEN GOSUB PQSUB2:
    price(scen) = price
    pnum(scen) = num
NEXT scen
RETURN
'***** END OF PQcompute SUBMODULE *****


PQSUB2:
'Submodule of PQcompute/PQsub4/PQsub3 submodule,
'   to compute prices and quantities
'   when fewer than 3 firms are producing.
num = 2
   quant(scen, num + 1) = 0
   pi(scen, num + 1) = 0
   costsum = 0
   FOR firm = 1 TO num
      costsum = costsum + cost(scen, firm)
   NEXT firm
   price = (A + costsum) / (num + 1)
   qtot = 0
   FOR firm = 1 TO num
      qtemp = (price - cost(scen, firm)) / b
      qtot = qtot + qtemp
      quant(scen, firm) = qtemp
      pitemp = (price - cost(scen, firm)) * qtemp
```

```
      pitemp = pitemp - fc
      pi(scen, firm) = pitemp
   NEXT firm
   cs(scen) = qtot * (A - price) / 2
   pdummy = 1 'Is last firm producing?
   IF quant(scen, num) < 0 THEN pdummy = 0
   IF pi(scen, num) < 0 THEN pdummy = 0
   IF pdummy = 0 THEN GOSUB PQSUB1:
RETURN
'***** END OF PQsub2SUBMODULE *****


PQSUB1:
'Submodule of PQcompute/PQsub4/PQsub3/PQsub2 submodule,
'   to compute prices and quantities
'   when fewer than 2 firms are producing.
num = 1
   quant(scen, num + 1) = 0
   pi(scen, num + 1) = 0
   costsum = 0
   FOR firm = 1 TO num
      costsum = costsum + cost(scen, firm)
   NEXT firm
   price = (A + costsum) / (num + 1)
   qtot = 0
   FOR firm = 1 TO num
      qtemp = (price - cost(scen, firm)) / b
      qtot = qtot + qtemp
      quant(scen, firm) = qtemp
      pitemp = (price - cost(scen, firm)) * qtemp
      pitemp = pitemp - fc
      pi(scen, firm) = pitemp
   NEXT firm
   cs(scen) = qtot * (A - price) / 2
   pdummy = 1 'Is last firm producing?
   IF quant(scen, num) < 0 THEN pdummy = 0
   IF pi(scen, num) < 0 THEN pdummy = 0
   IF pdummy = 0 THEN GOSUB PQSUB0:
RETURN
'***** END OF PQsub1 SUBMODULE *****


PQSUB0:
```

```
'Submodule of PQcompute/PQsub4,3,2,1 submodules,
'   to compute prices and quantities
'   when no firms are producing.
num = 0
   quant(scen, num + 1) = 0
   pi(scen, num + 1) = 0
   price = A
   cs(scen) = 0
RETURN
'***** END OF PQsub0 SUBMODULE *****


HHI:
'Submodule to compute Herfindahl-Herschmann Indices
'   for each given cost spread.
FOR scen = 0 TO 35
   qtot = 0
   FOR firm = 1 TO 3
      qtot = qtot + quant(scen, firm)
   NEXT firm
   IF qtot = 0 THEN
      HHI = 10000
      mshare(scen, 1) = 1
      mshare(scen, 2) = 0
      mshare(scen, 3) = 0
      GOTO 333
      END IF
   HHI = 0
   FOR firm = 1 TO 3
      mtemp = quant(scen, firm) / qtot
      mshare(scen, firm) = mtemp
      HHI = HHI + mtemp * mtemp * 10000
   NEXT firm
   herf(scen) = HHI
333 NEXT scen
RETURN
'***** END OF HHI SUBMODULE *****


PROFITS:
'This submodule assigns the previously computed
'   long-run business profits for each firm
'   to Microsoft and Microsoft's competitors.
```

```
'PiMS(s,f) = Microsoft's profit in scenario s,
'   assuming that Microsoft is firm f.
'PiComp(s,f) = Competitors' profits in scenario s,
'   assuming that Microsoft is firm f.
'If f=0, Microsoft has zero market share.
FOR scen = 0 TO 35
     pitot = 0
   FOR firm = 1 TO 3
     pitot = pitot + pi(scen, firm)
   NEXT firm
     pims(scen, 0) = 0
     picomp(scen, 0) = pitot
   FOR firm = 1 TO 3
     pitemp = pi(scen, firm)
     pitemp = pitemp * msfirms
     pims(scen, firm) = pitemp
     picomp(scen, firm) = pitot - pitemp
   NEXT firm
NEXT scen
'Microsoft profit (pitemp) is multiplied by the number of
Microsoft firms.
'When MSfirms=1, all profit calculations are accurate.
'When MSfirms=2 or 3, pims is accurate, but picomp is not
accurate for
'particular scenario/firm #, because this APM program
'does not simultaneously track more than one Microsoft firm.
'However, probability-weighted averages over all firm #s
'for a given scenario are an accurate average for both pims and
picomp.
RETURN
'***** END OF PROFITS SUBMODULE *****


PRINTTRAN0:
'This submodule prints the transition factors for
'   iteration zero, including average consumer surplus,
'   average profits for Microsoft and its competitors,
'   average market share for Microsoft, the industry-wide
'   Herfindahl-Hershman Index (HHI), and the average
'   number of main firms in the industry.
'These transition factors must be multiplied by Microsoft's
'   annual monopoly revenues to determine dollar values.
cstot = 0
```

```
pimstot = 0
picomptot = 0
mktshare = 0
herfindahl = 0
firmnum = 0
scen = 0 'Choose scen=scen0 to assume competitors in period 0.
   FOR firm = 0 TO 3
      tempprob = probl(0, firm)
      cstot = cstot + cs(scen) * tempprob
      pimstot = pimstot + pims(scen, firm) * tempprob
      picomptot = picomptot + picomp(scen, firm) * tempprob
      mktshare = mktshare + mshare(scen, firm) * tempprob
      herfindahl = herfindahl + herf(scen) * tempprob
      firmnum = firmnum + pnum(scen) * tempprob
   NEXT firm
PRINT #3, "Iter";
PRINT #3, " ConsumerSurpls ";
PRINT #3, "Profit(MS)     ";
PRINT #3, " Profit(comp)   ";
PRINT #3, " MktShare(1-MS)";
PRINT #3, " MktShare(n-MS)";
PRINT #3, " Herfindahl    ";
PRINT #3, " # firms       "
PRINT #3, USING "###"; 0;
PRINT #3, USING "###.############"; cstot; pimstot; picomptot;
PRINT #3, USING "####.##########"; mktshare * 100; msfirms *
mktshare * 100;
PRINT #3, USING "######.########"; herfindahl;
PRINT #3, USING "##.############"; firmnum
RETURN
'***** END OF PRINTTRAN0 SUBMODULE *****


PRINTTRAN:
'This submodule prints the transition factors for each
'   subsequent iteration, including average consumer surplus,
'   average profits for Microsoft and its competitors,
'   average market share for Microsoft, the industry-wide
'   Herfindahl-Hershman Index (HHI), and the average
'   number of main firms in the industry.
'These transition factors must be multiplied by Microsoft's
'   annual monopoly revenues to determine dollar values.
cstot = 0
```

```
pimstot = 0
picomptot = 0
mktshare = 0
herfindahl = 0
firmnum = 0
FOR scen = 1 TO 35
   FOR firm = 0 TO 3
      tempprob = probl(scen, firm)
      cstot = cstot + cs(scen) * tempprob
      pimstot = pimstot + pims(scen, firm) * tempprob
      picomptot = picomptot + picomp(scen, firm) * tempprob
      mktshare = mktshare + mshare(scen, firm) * tempprob
      herfindahl = herfindahl + herf(scen) * tempprob
      firmnum = firmnum + pnum(scen) * tempprob
   NEXT firm
NEXT scen
PRINT #3, USING "###"; iter;
PRINT #3, USING "###.############"; cstot; pimstot; picomptot;
PRINT #3, USING "####.##########"; mktshare * 100; msfirms *
mktshare * 100;
PRINT #3, USING "######.########"; herfindahl;
PRINT #3, USING "##.############"; firmnum
RETURN
'***** END OF PRINTTRAN SUBMODULE *****


PROBREAD0:
'Submodule to read iteration 0 transition probabilities.
LINE INPUT #2, temp$
INPUT #2, iter2, scen0
   IF 0 <> iter2 THEN PRINT "Iteration 0 mismatch:"; 0, iter2
FOR firm = 0 TO 3
   INPUT #2, probl(0, firm)
NEXT firm
RETURN
'***** END OF PROBREAD0 SUBMODULE *****


PROBREAD:
'Submodule to read subsequent iteration transition probabilities.
FOR scen = 1 TO 35
   INPUT #2, iter2, scen2
   IF iter <> iter2 THEN PRINT "Iteration S mismatch:"; iter,
```

```
iter2
   IF scen <> scen2 THEN PRINT "Scenario S mismatch:"; scen;
scen2
   FOR firm = 0 TO 3
      INPUT #2, probl(scen, firm)
   NEXT firm
NEXT scen
RETURN
'***** END OF PROBREAD SUBMODULE *****

'**********END OF Program "MS4TranA.bas".**********
```

Attachment P.

```
'BASIC Program "MS5TranR.bas".

'Program Number 5 in a series of six programs
'designed to simulate alternative antitrust
'remedies for the Microsoft software industry.

'Copyright, January 23, 2002, Carl Lundgren.

'   This program, "MS5TranR.bas", uses the probability data
'computed by "MS3ProbR.bas" to compute Consumer Surplus and
'Profits for both Microsoft and Microsoft's competitors.
'In transition period zero (iter=0), Microsoft (and its
'successor firms after divestiture) are assumed to have no
competitors.
'In subsequent transition periods (iter=1 to 10),
'Microsoft has (potentially) one or more competitors.
'This program calculates both Relative Profit Maximizing ("RPM")
'and Absolute Profit Maximizing ("APM") equilibria.

'   The program uses the computed probabilities for each
'scenario that was previously outputted by the
'"MS3ProbR.bas" program as various "PROB....txt" files.
'   This program outputs as "TRPM....txt" files the
'computed transition factors for several alternative timepaths
'for the software industry, under several alternative
assumptions.
'These transition factors are computed as a fraction
'of the revenues which Microsoft would earn if it remained
'a monopoly.
'   This program computes transition factors
'for alternative timepaths for the software industry,
'under the assumption that Microsoft is split into two firms,
'and these two firms adopt relative profit maximizing (RPM)
'incentives in either a pure or impure form.
'The goal functions for the two RPM firms are:
'          Goal1 = Profit1 - z * Profit2
'          Goal2 = Profit2 - z * Profit1
'All other (non-Microsoft, competitor) firms are assumed
'to have absolute profit maximizing (APM) incentives.

'The assumed values for z in the transitions are:
'    TRPM0) The value of z=0.0 'Same as Absolute Profit
```

```
Maximizing (APM).
'     TRPM1) The value of z=0.1 '10% RPM, 90% APM.
'     TRPM2) The value of z=0.2 '20% RPM, 80% APM.
'     TRPM3) The value of z=0.3 '30% RPM, 70% APM.
'     TRPM4) The value of z=0.4 '40% RPM, 60% APM.
'     TRPM5) The value of z=0.5 '50% RPM, 50% APM.
'     TRPM6) The value of z=0.6 '60% RPM, 40% APM.
'     TRPM7) The value of z=0.7 '70% RPM, 30% APM.
'     TRPM8) The value of z=0.8 '80% RPM, 20% APM.
'     TRPM9) The value of z=0.9 '90% RPM, 10% APM.

'   This program differs from the MS4TranA.bas program
'in that it considers only two successor firms for Microsoft,
'and simultaneously tracks the rankings of both firms.
'   The 750 outputted transition (TRPM....txt) files are computed
'for three speeds of transition (speed=1,2,3),
'five cost ratios for short-run cost (cratio=1,2,3,4,5),
'five assumptions concerning the portion of long-run costs
'allocated to fixed costs (port=0,1,2,3,4),
'ten different values of z
'(z = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9).
'The starting point for the transitions in this program
'always has Microsoft divided into two RPM firms (msfirms=2).
'   An additional feature of the program allows the value of z to
'change in response to circumstances.  If zbump=0.0, then z is
fixed,
'and does not change in response to circumstances.  If zbump > 0,
'then z changes in response to circumstances.  In the program,
'z responds to the circumstance that one of the RPM firms
'is not producing, because it is achieving negative absolute
profit.
'In this circumstance, the program automatically "bumps down" the
value
'of z for both RPM firms by the amount of zbump.  For example,
'if z=0.7 and zbump=0.3, then if one or both RPM firms would shut
down,
'then the value of z is automatically bumped down to z=0.4.
'In many circumstances, this allows both RPM firms to continue
producing.
'   The user determines the value of zbump as part of the control
module.


DEFDBL A-Z
```

```
DIM problr(35, 15), herf(35, 15), sharems(35, 15), pnum(35, 15)
DIM qtotal(35, 15), cost(35, 3), pv(35, 3), price(35, 15)
DIM pims(35, 15), picomp(35, 15)
DIM cs(35, 15), pi(35, 15)


'CONTROL MODULE
CLS
timex = TIMER
'*****User Determines amount by which z should be bumped down,
'     if RPM firm2 is not producing when z=zhold(zcount).*****
zbump = 0!   '*****User determines zbump.*****
'*****If zbump=0, then z is fixed and never changes.
'*****zbump >= 0.  Recommended value is zbump=0.3.****
GOSUB PRINTZCOUNT:
GOSUB SCENREAD:
'This section calls the main module 750 times.
'This control module chooses market tendency:
   'cratio=1 'Ratio for low/high short-run cost is 0.2500
(1/4.0).
   'cratio=2 'Ratio for low/high short-run cost is 0.3333
(1/3.0).
   'cratio=3 'Ratio for low/high short-run cost is 0.4000
(1/2.5).
   'cratio=4 'Ratio for low/high short-run cost is 0.5000
(1/2.0).
   'cratio=5 'Ratio for low/high short-run cost is 0.6667
(1/1.5).
'This control module chooses speed for market share shifts:
   'speed = 1 'Slow speed for market share shifts.
   'speed = 2 'Moderate speed for market share shifts.
   'speed = 3 'High speed for market share shifts.
'In this program, # of msfirms at iteration zero is always two.
msfirms = 2
'This control module chooses z (weight on rival firm's profits).
   'zcount=0 'z = 0.0
   'zcount=1 'z = 0.1
   'zcount=2 'z = 0.2
   'zcount=3 'z = 0.3
   'zcount=4 'z = 0.4
   'zcount=5 'z = 0.5
   'zcount=6 'z = 0.6
   'zcount=7 'z = 0.7
   'zcount=8 'z = 0.8
```

```
    'zcount=9 'z = 0.9
'This control module chooses proportion of long-run cost
'   which is assumed to be a fixed cost.
    'port = 0 'Fixed cost is 0% of long-run cost.
    'port = 1 'Fixed cost is 25% of long-run cost.
    'port = 2 'Fixed cost is 50% of long-run cost.
    'port = 3 'Fixed cost is 75% of long-run cost.
    'port = 4 'Fixed cost is 100% of long-run cost.

FOR cratio = 1 TO 5
   FOR speed = 1 TO 3
      FOR port = 0 TO 4
         FOR zcount = 0 TO 9
            GOSUB MAINMODULE:
         NEXT zcount
      NEXT port
   NEXT speed
NEXT cratio
PRINT TIMER - timex
END


PRINTZCOUNT:
'This submodule assigns values of z to each zcount,
'   and prints these z values for transfer to
'   the subsequent "MS6Summ.bas" program.
DIM zhold(9)
zcount$ = "c:\basic\ms_sim\out\zcount.txt" 'Output zcount data.
OPEN zcount$ FOR OUTPUT AS #1
PRINT #1, "Zcount  Z"
FOR zcount = 0 TO 9
   IF zcount = 0 THEN z = 0!
   IF zcount = 1 THEN z = .1
   IF zcount = 2 THEN z = .2
   IF zcount = 3 THEN z = .3
   IF zcount = 4 THEN z = .4
   IF zcount = 5 THEN z = .5
   IF zcount = 6 THEN z = .6
   IF zcount = 7 THEN z = .7
   IF zcount = 8 THEN z = .8
   IF zcount = 9 THEN z = .9
   zhold(zcount) = z
   PRINT #1, USING "######"; zcount;
   PRINT #1, USING "###.######"; z
```

```
NEXT zcount
CLOSE #1
RETURN
'***** END OF PrintZcount SUBMODULE *****


MAINMODULE:
z = zhold(zcount)
GOSUB FILENAMES: 'Assign file names to input/output files.
GOSUB INITIALIZE:
FOR iter = 1 TO 10
   GOSUB PROBREAD:
   GOSUB PRINTTRAN:
NEXT iter
CLOSE
RETURN
'***** END OF MAIN MODULE *****


FILENAMES:
prob$ = "c:\basic\ms_sim\out\probr0.txt" 'Input RPM transition
probabilities
tran$ = "c:\basic\ms_sim\out\tRPM0000.txt" 'Output RPM transition
factors
IF cratio = 1 THEN crt$ = "1"
IF cratio = 2 THEN crt$ = "2"
IF cratio = 3 THEN crt$ = "3"
IF cratio = 4 THEN crt$ = "4"
IF cratio = 5 THEN crt$ = "5"
IF speed = 1 THEN sp$ = "1"
IF speed = 2 THEN sp$ = "2"
IF speed = 3 THEN sp$ = "3"
IF zcount = 0 THEN zc$ = "0"
IF zcount = 1 THEN zc$ = "1"
IF zcount = 2 THEN zc$ = "2"
IF zcount = 3 THEN zc$ = "3"
IF zcount = 4 THEN zc$ = "4"
IF zcount = 5 THEN zc$ = "5"
IF zcount = 6 THEN zc$ = "6"
IF zcount = 7 THEN zc$ = "7"
IF zcount = 8 THEN zc$ = "8"
IF zcount = 9 THEN zc$ = "9"
IF port = 0 THEN prt$ = "0"
IF port = 1 THEN prt$ = "1"
```

```
IF port = 2 THEN prt$ = "2"
IF port = 3 THEN prt$ = "3"
IF port = 4 THEN prt$ = "4"
replacep$ = sp$
replacet$ = zc$ + crt$ + sp$ + prt$
MID$(prob$, 26, 1) = replacep$
MID$(tran$, 25, 4) = replacet$
PRINT replacet$; " ";
RETURN
'***** END OF FileNames SUBMODULE *****


INITIALIZE:
'Submodule to perform various initialization tasks.
OPEN prob$ FOR INPUT AS #2
OPEN tran$ FOR OUTPUT AS #3
GOSUB ZEROPROB:
GOSUB PROBREAD0:
GOSUB SCENREAD: 'Read scenario list.
GOSUB COSTCOMPUTE: 'Compute costs.
GOSUB PQZERO: 'Iteration 0 prices, quantities, profits & Consumer
Surplus.
GOSUB PQCOMPUTE: 'Compute prices, quantities, profits & Consumer
Surplus.
GOSUB PRINTTRAN0: 'Print transition files.
RETURN
'***** END OF Initialize SUBMODULE *****


ZEROPROB:
'This submodule sets the prob1r(0, .)
'   variable values to zero.
FOR pair = 0 TO 15
   prob1r(0, pair) = 0
NEXT pair
RETURN
'***** END OF ZeroProb SUBMODULE *****


SCENREAD:
'This submodule reads in the scenario costs list.
cost$ = "c:\basic\ms_sim\costlist.txt" 'Input scenario costs
OPEN cost$ FOR INPUT AS #1
LINE INPUT #1, dummy$
```

```
'cost(s,f) = short-run marginal cost of firm f in scenario s.
'finprob(s) = final probability assumed for scenario s.
'wgt(scen) = number of permutations of scenario s.
FOR scen = 1 TO 35
   INPUT #1, scen2, L1, L2, L3, L4, L5, wgt, finprob
     IF scen <> scen2 THEN PRINT "Scenario mismatch", scen, scen2
   INPUT #1, c1, c2, c3
   FOR firm = 1 TO 3
     INPUT #1, pv(scen, firm)
   NEXT firm
NEXT scen
CLOSE #1
RETURN
'***** END OF ScenRead SUBMODULE *****


COSTCOMPUTE:
'Submodule to compute short-run costs, long-run costs,
'   and assumed elasticity of demand.

'This section computes parameters for long-run costs
'   under the assumption that each firm has
'   the same long-run cost function.
'Assume that one portion of Microsoft's
'   long-run cost (LRC) is a long-run fixed cost (FC),
'   while the other portion is a long-run variable cost (VC),
'   which is proportional to output.
lrc = .1855 'computed as MS long-run cost divided by MS monopoly
revenue.
IF port = 0 THEN portion = 0!
IF port = 1 THEN portion = .25
IF port = 2 THEN portion = .5
IF port = 3 THEN portion = .75
IF port = 4 THEN portion = 1!
fc = lrc * portion
vc = lrc * (1 - portion)

'This section computes elasticity of demand (Elas) at
'   monopoly profit maximum, as a function of marginal cost,
'   which is composed of short-run marginal cost (SRC)
'   plus long-run variable cost (VC).
src = .4101 'computed as MS short-run cost divided by MS monopoly
revenue.
mc = src + vc
```

```
elas = 1 / (mc - 1)
elasminus = elas - 1
elasplus = elas + 1
A = elasminus / elas 'Intercept of linear demand curve with price
axis.
b = -1 / elas 'Slope of linear demand curve.
cbase = src 'Base level of short-run marginal cost (cost level
2).
logcbase = LOG(cbase / (A - vc - cbase)) 'Cbase converted to log-
ratios.

'This section computes short-run costs and marginal costs
'   for a given cost spread.
IF cratio = 1 THEN cspread = .950980935#
IF cratio = 2 THEN cspread = .748669813#
IF cratio = 3 THEN cspread = .622288438#
IF cratio = 4 THEN cspread = .469161475#
IF cratio = 5 THEN cspread = .273626703#
FOR scen = 1 TO 35
   FOR firm = 1 TO 3
      pvtemp = (pv(scen, firm) - 180) / 120#
      logpv = logcbase + pvtemp * cspread
      pvratio = EXP(logpv)
      cost(scen, firm) = vc + (A - vc) * pvratio / (1 + pvratio)
   NEXT firm
NEXT scen
RETURN
'***** END OF CostCompute SUBMODULE *****


PQZERO:
'Submodule to compute prices, quantities, profits,
'   and consumer surplus for selected scenarios,
'   for iteration zero, where two Microsoft RPM firms
'   are assumed initially to have no competitors.
'This program assumes that the two MS firms use
'   relative profit maximizing (RPM) incentives,
'   according to the goal functions for each firm:
'   Goal(firm1)=profit(firm1)-z*profit(firm2)
'   Goal(firm2)=profit(firm2)-z*profit(firm1)
scen = 0
firm1 = 1
firm2 = 2
pair1 = firm1 * 4 + firm2
```

```
pair2 = firm2 * 4 + firm1
FOR firm = 1 TO 2
   cost(0, firm) = cost(scen0, firm)
NEXT firm
FOR firm = 3 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
   cost(0, firm) = A
NEXT firm
bump = 0 'Dummy variable to determine if z should be bumped down.
z = zhold(zcount)
GOSUB RPMSUB0:
IF bump = 1 THEN
   z = z - zbump
   GOSUB RPMSUB0:
   END IF
GOSUB ASSIGN:
RETURN
'***** END OF PQzero SUBMODULE *****


PQCOMPUTE:
'Submodule to compute prices, quantities, profits,
'   and consumer surplus for all scenarios and firm pairs.
FOR scen = 1 TO 35
  FOR firm1 = 0 TO 3
      firm2 = 0
      pair1 = firm1 * 4 + firm2
      pair2 = firm2 * 4 + firm1
      delfirm = 0
      bump = 0 'Dummy variable to determine if z should be bumped
down.
      z = zhold(zcount)
      GOSUB APMCOMPUTE:
      GOSUB ASSIGN:
  NEXT firm1
  FOR firm1 = 1 TO 3
    FOR firm2 = 1 TO 3
      IF firm2 <= firm1 THEN 357
      pair1 = firm1 * 4 + firm2
      pair2 = firm2 * 4 + firm1
      bump = 0 'Dummy variable to determine if z should be bumped
down.
      z = zhold(zcount)
```

```
      GOSUB RPMCOMPUTE:
      IF bump = 1 THEN
         z = z - zbump
         GOSUB RPMCOMPUTE:
         END IF
      GOSUB ASSIGN:
357 NEXT firm2
  NEXT firm1
NEXT scen
RETURN
'***** END OF PQcompute SUBMODULE *****


ASSIGN:
'Submodule to assign computed numbers for each scenario
'   and combination of firms.
qtotal(scen, pair1) = qtot
price(scen, pair1) = price
pnum(scen, pair1) = tnum
cs(scen, pair1) = cstemp
qtotal(scen, pair2) = qtot
price(scen, pair2) = price
pnum(scen, pair2) = tnum
cs(scen, pair2) = cstemp
IF qtot = 0 THEN
   share(1) = 1
   share(2) = 0
   share(3) = 0
   GOTO 333
   END IF
FOR firm = 1 TO 3
   share(firm) = qtemp(firm) / qtot
NEXT firm
333 pitot = 0
herf = 0
FOR firm = 1 TO 3
   pitot = pitot + pitemp(firm)
   herf = herf + share(firm) * share(firm)
NEXT firm
herf = herf * 10000
herf(scen, pair1) = herf
herf(scen, pair2) = herf
ms1share = share(firm1)
ms2share = share(firm2)
```

```
msavgshare = (ms1share + ms2share) / 2
sharems(scen, pair1) = msavgshare
sharems(scen, pair2) = msavgshare
mspi1 = pitemp(firm1)
mspi2 = pitemp(firm2)
IF firm1 = 0 THEN mspi1 = 0
IF firm2 = 0 THEN mspi2 = 0
mspitot = mspi1 + mspi2
comppitot = pitot - mspitot
pims(scen, pair1) = mspitot
pims(scen, pair2) = mspitot
picomp(scen, pair1) = comppitot
picomp(scen, pair2) = comppitot
ms1goal = mspi1 - z * mspi2 'Firm 1's RPM goal function.
ms2goal = mspi2 - z * mspi1 'Firm 2's RPM goal function.
RETURN
'***** END OF Assign SUBMODULE *****


RPMCOMPUTE:
'Submodule to compute prices, quantities, profits,
'   and consumer surplus for each RPM scenario.
'   This submodule assumes that two RPM firms
'   choose to produce.
anum = 1 'anum = number of producing APM firms.
num = anum 'num = last producing APM firm.
IF num = firm1 THEN num = num + 1
IF num = firm2 THEN num = num + 1
IF num = firm1 THEN num = num + 1
tnum = anum + 2 'tnum = total number of producing firms.
FOR firm = 1 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
NEXT firm
   costsum = 0
   FOR firm = 1 TO num
      IF firm <> firm1 AND firm <> firm2 THEN
         costsum = costsum + cost(scen, firm)
      END IF
   NEXT firm
   costsum = (A + costsum) * (1 - z)
   costsum = costsum + cost(scen, firm1) + cost(scen, firm2)
   price = costsum / (3 - anum * z + anum - z)
   qtot = (A - price) / b
```

```
   qapm = 0
   FOR firm = 1 TO num
      IF firm <> firm1 AND firm <> firm2 THEN
         qtemp(firm) = (price - cost(scen, firm)) / b
         qapm = qapm + qtemp(firm)
         pitemp = (price - cost(scen, firm)) * qtemp(firm)
         pitemp(firm) = pitemp - fc
      END IF
   NEXT firm
   qrpm = qtot - qapm
   qgap = (cost(scen, firm2) - cost(scen, firm1)) / b / (1 + z)
   qtemp(firm1) = (qrpm + qgap) / 2
      pitemp = (price - cost(scen, firm1)) * qtemp(firm1)
      pitemp(firm1) = pitemp - fc
   qtemp(firm2) = (qrpm - qgap) / 2
      pitemp = (price - cost(scen, firm2)) * qtemp(firm2)
      pitemp(firm2) = pitemp - fc
   cstemp = qtot * (A - price) / 2
'Must choose which firm (if any) to shut down,
'   based on quantities and profits.
'First, test for negative quantities.
aq = 1 'quantity dummy for APM firm.
   IF qtemp(num) < 0 THEN aq = 0
rq = 1 'quantity dummy for RPM firm.
   IF qtemp(firm2) < 0 THEN rq = 0
IF aq = 0 AND rq = 1 THEN
   GOSUB RPMSUB0:
   RETURN
   END IF
IF aq = 1 AND rq = 0 THEN
   IF bump = 0 THEN
      bump = 1
      RETURN
      END IF
   delfirm = firm2
   GOSUB APMCOMPUTE:
   RETURN
   END IF
IF aq = 0 AND rq = 0 THEN
   IF firm2 > num THEN
      IF bump = 0 THEN
         bump = 1
         RETURN
         END IF
```

```
        delfirm = firm2
        GOSUB APMCOMPUTE:
        RETURN
        END IF
     IF firm2 < num THEN
        GOSUB RPMSUB0:
        RETURN
        END IF
     END IF
'Second, test for negative profits.
api = 1 'profit dummy for APM firm.
   IF pitemp(num) < 0 THEN api = 0
rpi = 1 'profit dummy for RPM firm.
   IF pitemp(firm2) < 0 THEN rpi = 0
IF api = 0 AND rpi = 1 THEN
   GOSUB RPMSUB0:
   RETURN
   END IF
IF api = 1 AND rpi = 0 THEN
   IF bump = 0 THEN
      bump = 1
      RETURN
      END IF
   delfirm = firm2
   GOSUB APMCOMPUTE:
   RETURN
   END IF
IF api = 0 AND rpi = 0 THEN
   IF pitemp(firm2) < pitemp(num) THEN
      IF bump = 0 THEN
         bump = 1
         RETURN
         END IF
      delfirm = firm2
      GOSUB APMCOMPUTE:
      RETURN
   ELSE
      GOSUB RPMSUB0:
      RETURN
   END IF
END IF
'If program reaches here, then all firms are producing.
RETURN
'***** END OF RPMcompute SUBMODULE *****
```

```
RPMSUB0:
'Submodule of RPMcompute submodule,
'   to compute prices and quantities
'   when zero APM firms are producing.
num = 0
anum = num 'anum = number of producing APM firms
tnum = anum + 2 'tnum = total number of producing firms
FOR firm = 1 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
NEXT firm
   qtemp(num + 1) = 0
   pitemp(num + 1) = 0
   costsum = 0
   costsum = (A + costsum) * (1 - z)
   costsum = costsum + cost(scen, firm1) + cost(scen, firm2)
   price = costsum / (3 - anum * z + anum - z)
   qtot = (A - price) / b
   qapm = 0
   qrpm = qtot - qapm
   qgap = (cost(scen, firm2) - cost(scen, firm1)) / b / (1 + z)
   qtemp(firm1) = (qrpm + qgap) / 2
      pitemp = (price - cost(scen, firm1)) * qtemp(firm1)
      pitemp(firm1) = pitemp - fc
   qtemp(firm2) = (qrpm - qgap) / 2
      pitemp = (price - cost(scen, firm2)) * qtemp(firm2)
      pitemp(firm2) = pitemp - fc
   cstemp = qtot * (A - price) / 2
r2dummy = 1 'Is RPM firm2 producing?
   IF qtemp(firm2) < 0 THEN r2dummy = 0
   IF pitemp(firm2) < 0 THEN r2dummy = 0
IF r2dummy = 0 THEN
   IF firm1 = 1 THEN
      IF bump = 0 THEN
         bump = 1
         RETURN
         END IF
      delfirm = 0
      GOSUB APMSUB1:
      RETURN
      END IF
   IF firm1 = 2 THEN
      IF bump = 0 THEN
```

```
          bump = 1
          RETURN
          END IF
       delfirm = 1
       GOSUB APMSUB2:
       RETURN
       END IF
    END IF
'If program reaches here, then both RPM firms are producing.
RETURN
'***** END OF RPMsub0 SUBMODULE *****


APMCOMPUTE:
'   Submodule to compute prices, quantities, profits,
'and consumer surplus for each APM scenario.
'The delfirm variable is used to determine whether to
'delete one of the firms from the APM scenario.
'  If delfirm=0, no firms are deleted from the computation.
'  If delfirm=1, then firm 1 is deleted from the computation.
'  If delfirm=2, then firm 2 is deleted from the computation.
'  If delfirm=3, then firm 3 is deleted from the computation.
num = 3
IF delfirm = num THEN GOTO APMSUB2:
anum = num 'anum = number of producing APM firms.
IF delfirm > 0 THEN
   IF delfirm <= num THEN anum = num - 1
END IF
tnum = anum 'tnum = total number of producing firms
FOR firm = 1 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
NEXT firm
   costsum = 0
   FOR firm = 1 TO num
      IF firm <> delfirm THEN
      costsum = costsum + cost(scen, firm)
      END IF
   NEXT firm
   price = (A + costsum) / (anum + 1)
   qtot = 0
   FOR firm = 1 TO num
      IF firm <> delfirm THEN
        qtemp(firm) = (price - cost(scen, firm)) / b
```

```
        qtot = qtot + qtemp(firm)
        pitemp = (price - cost(scen, firm)) * qtemp(firm)
        pitemp(firm) = pitemp - fc
        END IF
      IF firm = delfirm THEN
        qtemp(firm) = 0
        pitemp(firm) = 0
        END IF
   NEXT firm
   cstemp = qtot * (A - price) / 2
   pdummy = 1 'Is last firm producing?
   IF qtemp(num) < 0 THEN pdummy = 0
   IF pitemp(num) < 0 THEN pdummy = 0
   IF pdummy = 0 THEN GOSUB APMSUB2:
RETURN
'***** END OF APMcompute SUBMODULE *****


APMSUB2:
'Submodule of APMcompute submodule,
'   to compute prices and quantities
'   when fewer than 3 firms are producing.
num = 2
IF delfirm = num THEN GOTO APMSUB1:
anum = num 'anum = number of producing APM firms.
IF delfirm > 0 THEN
   IF delfirm <= num THEN anum = num - 1
END IF
tnum = anum 'tnum = total number of producing firms
FOR firm = 1 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
NEXT firm
   costsum = 0
   FOR firm = 1 TO num
      IF firm <> delfirm THEN
      costsum = costsum + cost(scen, firm)
      END IF
   NEXT firm
   price = (A + costsum) / (anum + 1)
   qtot = 0
   FOR firm = 1 TO num
      IF firm <> delfirm THEN
        qtemp(firm) = (price - cost(scen, firm)) / b
```

```
          qtot = qtot + qtemp(firm)
          pitemp = (price - cost(scen, firm)) * qtemp(firm)
          pitemp(firm) = pitemp - fc
          END IF
        IF firm = delfirm THEN
          qtemp(firm) = 0
          pitemp(firm) = 0
          END IF
    NEXT firm
    cstemp = qtot * (A - price) / 2
    pdummy = 1 'Is last firm producing?
    IF qtemp(num) < 0 THEN pdummy = 0
    IF pitemp(num) < 0 THEN pdummy = 0
    IF pdummy = 0 THEN GOSUB APMSUB1:
RETURN
'***** END OF APMsub2SUBMODULE *****


APMSUB1:
'Submodule of APMcompute/APMsub2 submodules,
'   to compute prices and quantities
'   when fewer than 2 firms are producing.
num = 1
IF delfirm = num THEN GOTO APMSUB0:
anum = num 'anum = number of producing APM firms.
IF delfirm > 0 THEN
   IF delfirm <= num THEN anum = num - 1
END IF
tnum = anum 'tnum = total number of producing firms
FOR firm = 1 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
NEXT firm
   costsum = 0
   FOR firm = 1 TO num
      IF firm <> delfirm THEN
      costsum = costsum + cost(scen, firm)
      END IF
   NEXT firm
   price = (A + costsum) / (anum + 1)
   qtot = 0
   FOR firm = 1 TO num
      IF firm <> delfirm THEN
        qtemp(firm) = (price - cost(scen, firm)) / b
```

```
          qtot = qtot + qtemp(firm)
          pitemp = (price - cost(scen, firm)) * qtemp(firm)
          pitemp(firm) = pitemp - fc
          END IF
        IF firm = delfirm THEN
          qtemp(firm) = 0
          pitemp(firm) = 0
          END IF
    NEXT firm
    cstemp = qtot * (A - price) / 2
    pdummy = 1 'Is last firm producing?
    IF qtemp(num) < 0 THEN pdummy = 0
    IF pitemp(num) < 0 THEN pdummy = 0
    IF pdummy = 0 THEN GOSUB APMSUB0:
RETURN
'***** END OF APMsub1 SUBMODULE *****


APMSUB0:
'Submodule of APMcompute/APMsub2,1 submodules,
'   to compute prices and quantities
'   when no firms are producing.
num = 0
anum = num 'anum = number of producing APM firms.
tnum = anum 'tnum = total number of producing firms
FOR firm = 1 TO 3
   qtemp(firm) = 0
   pitemp(firm) = 0
NEXT firm
   price = A
   cstemp = 0
RETURN
'***** END OF APMsub0 SUBMODULE *****


PRINTTRAN0:
'This submodule prints the transition factors for
'   iteration zero, including average consumer surplus,
'   average profits for Microsoft and its competitors,
'   average market share for Microsoft, the industry-wide
'   Herfindahl-Hershman Index (HHI), and the average
'   number of main firms in the industry.
'These transition factors must be multiplied by Microsoft's
'   annual monopoly revenues to determine dollar values.
```

```
cstot = 0
pimstot = 0
picomptot = 0
mktshare = 0
herfindahl = 0
firmnum = 0
scen = 0 'Choose scen=scen0 to assume competitors in period 0.
   FOR firm1 = 0 TO 3
      FOR firm2 = 0 TO 3
         pair = 4 * firm1 + firm2
         tempprob = prob1r(scen, pair)
         cstot = cstot + cs(scen, pair) * tempprob
         pimstot = pimstot + pims(scen, pair) * tempprob
         picomptot = picomptot + picomp(scen, pair) * tempprob
         mktshare = mktshare + sharems(scen, pair) * tempprob
         herfindahl = herfindahl + herf(scen, pair) * tempprob
         firmnum = firmnum + pnum(scen, pair) * tempprob
      NEXT firm2
   NEXT firm1
PRINT #3, "Iter";
PRINT #3, " ConsumerSurpls ";
PRINT #3, "Profit(MS)      ";
PRINT #3, " Profit(comp)    ";
PRINT #3, " MktShare(1-MS)";
PRINT #3, " MktShare(n-MS)";
PRINT #3, " Herfindahl     ";
PRINT #3, " # firms        "
PRINT #3, USING "###"; 0;
PRINT #3, USING "###.############"; cstot; pimstot; picomptot;
PRINT #3, USING "####.##########"; mktshare * 100; msfirms *
mktshare * 100;
PRINT #3, USING "######.########"; herfindahl;
PRINT #3, USING "##.############"; firmnum
RETURN
'***** END OF PrintTran0 SUBMODULE *****


PRINTTRAN:
'This submodule prints the transition factors for each
'   subsequent iteration, including average consumer surplus,
'   average profits for Microsoft and its competitors,
'   average market share for Microsoft, the industry-wide
'   Herfindahl-Hershman Index (HHI), and the average
'   number of main firms in the industry.
```

```
'These transition factors must be multiplied by Microsoft's
'   annual monopoly revenues to determine true dollar values.
cstot = 0
pimstot = 0
picomptot = 0
mktshare = 0
herfindahl = 0
firmnum = 0
FOR scen = 1 TO 35
   FOR firm1 = 0 TO 3
      FOR firm2 = 0 TO 3
         pair = 4 * firm1 + firm2
         tempprob = prob1r(scen, pair)
         cstot = cstot + cs(scen, pair) * tempprob
         pimstot = pimstot + pims(scen, pair) * tempprob
         picomptot = picomptot + picomp(scen, pair) * tempprob
         mktshare = mktshare + sharems(scen, pair) * tempprob
         herfindahl = herfindahl + herf(scen, pair) * tempprob
         firmnum = firmnum + pnum(scen, pair) * tempprob
      NEXT firm2
   NEXT firm1
NEXT scen
PRINT #3, USING "###"; iter;
PRINT #3, USING "###.############"; cstot; pimstot; picomptot;
PRINT #3, USING "####.##########"; mktshare * 100; msfirms *
mktshare * 100;
PRINT #3, USING "######.########"; herfindahl;
PRINT #3, USING "##.############"; firmnum
RETURN
'***** END OF PrintTran SUBMODULE *****


PROBREAD0:
LINE INPUT #2, temp$
FOR firm1 = 0 TO 3
   INPUT #2, iter2, scen0, firm
      IF 0 <> iter2 THEN PRINT "Iteration mismatch:"; 0; iter2
      IF firm <> firm1 THEN PRINT "Firm1 mismatch:"; firm1; firm
   FOR firm2 = 0 TO 3
      pair = firm1 * 4 + firm2
      INPUT #2, prob1r(0, pair)
   NEXT firm2
NEXT firm1
RETURN
```

```
'***** END OF ProbRead0 SUBMODULE *****


PROBREAD:
FOR scen = 1 TO 35
FOR firm1 = 0 TO 3
   INPUT #2, iter2, scen2, firm
      IF iter <> iter2 THEN PRINT "Iteration mismatch:"; iter;
iter2
      IF scen <> scen2 THEN PRINT "Scenario mismatch:"; scen;
scen2
      IF firm <> firm1 THEN PRINT "Firm1 mismatch:"; firm1; firm
   FOR firm2 = 0 TO 3
      pair = firm1 * 4 + firm2
      INPUT #2, prob1r(scen, pair)
   NEXT firm2
NEXT firm1
NEXT scen
RETURN
'***** END OF ProbRead SUBMODULE *****

'**********END OF Program "MS5TranR.bas".**********
```

Attachment Q.

```
'BASIC Program "MS6Summ.bas".

'Program Number 6 in a series of six programs
'designed to simulate alternative antitrust
'remedies for the Microsoft software industry.

'Copyright, January 23, 2002, Carl Lundgren.

'   This program, "MS6Summ.bas", computes and summarizes the data
'produced by prior programs, including both "MS4TranA.bas"
'and "MS5TranR.bas".

'   This program summarizes in the form
'of aggregates and comparisons the economic meaning
'of the transitions data that were outputted by
'the "MS4TranA.bas and "MS5TranR.bas" programs.
'This program reads in the various "TRAN....txt"
'and "TRPM....txt" files produced by the prior programs
'in order to create summary files for the aggregates:
'   0) No remedy at all, continued monopoly by Microsoft.
'   1) 100% effective conduct remedy starting in 2002.
'   1a) Intermediate conduct remedies, varying
'       in effectiveness: 20%, 40%, 60%, 80%.
'   1b) Structural two-monopolies remedy, computed as
'       having outcomes equivalent to one-third value of
'       2-firm competitive APM structural remedy and
'       two-thirds value of 100% conduct remedy.
'   2) Structural 2-firm APM remedy starting in 2005.
'   3) Structural 3-firm APM remedy starting in 2005.
'   4-15) Structural 2-firm RPM remedies for z=0.1 through z=0.9,
'         starting in 2005.
'   16) Lawful behavior since 1995.
'and for the comparisons:
'   17) Aggregates for all the above alternatives,
'       minus the aggregates for the lawful path.
'
'   The program uses the transition data to estimate
'consumer surplus and profits in each of the years 1995-2025.
'Transitions are assumed to take place over a period
'of 3, 5, or 8 years each.  Years in between the transition
'years are linearly interpolated.
'    These four time paths are aggregated and
```

```
'compared for three sums over three time periods:
'   1) Sum of consumer surpluses.
'   2) Sum of non-Microsoft profits.
'   3) Sum of Microsoft profits.
'   4) Sum of total surpluses.
'   A) Time period 1995-2001.
'   B) Time period 2002-2025.
'   C) Time period 1995-2025.

'    The program reads in data for Microsoft's monopoly
' revenues by year, multiplies them by the relevant factor
' multipliers given by the Transition files (TRAN & TRPM),
' and computes interest or discounts at 7% real annual
' interest rate to billions of year 2002 real dollars.
'   The program produces data summarized for particular scenarios
'in files marked "AGGC....txt", "AGGR....txt", and "YEAR....txt".
'The "AGGC....txt" files (which are most user friendly) summarize
'all past and future data, appropriately discounted, into a
single
'set of figures which may be compared across remedy proposals.
'The "AGGR....txt" files  categorize the aggregate data into
'past and future amounts of consumer surplus, profits, and
'total surplus for each remedy proposal, and how these amounts
'compare with the same amounts along the lawful path.
'The "YEAR....txt" files (which are least user friendly) output
'the calculated amounts, by year, for each remedy proposal and
'the lawful path.

DEFDBL A-Z
DIM aggcs(16, 3), aggcomp(16, 3), aggms(16, 3), aggts(16, 3)
'Aggregates
DIM compcs(16, 3), compcomp(16, 3), compms(16, 3), compts(16, 3)
'Comparisons
DIM aggcsmin(16, 3), aggcompmin(16, 3), aggmsmin(16, 3),
aggtsmin(16, 3) 'Minimums
DIM compcsmin(16, 3), compcompmin(16, 3), compmsmin(16, 3),
comptsmin(16, 3) 'Minimums
DIM aggcsmax(16, 3), aggcompmax(16, 3), aggmsmax(16, 3),
aggtsmax(16, 3) 'Maximums
DIM compcsmax(16, 3), compcompmax(16, 3), compmsmax(16, 3),
comptsmax(16, 3) 'Minimums
DIM aggcsavg(16, 3), aggcompavg(16, 3), aggmsavg(16, 3),
aggtsavg(16, 3) 'Averages
DIM compcsavg(16, 3), compcompavg(16, 3), compmsavg(16, 3),
```

```
IF port >= 0 THEN 'Always true; change to restrict statistics
gathering.
   pvtotal = pvtotal + pvtemp
   avgtotal = avgtotal + 1
FOR p = 0 TO 14
FOR t = 1 TO 3
'First put numbers into temporary variable slots.
   aggcstemp = aggcs(p, t)
   aggcomptemp = aggcomp(p, t)
   aggmstemp = aggms(p, t)
   aggtstemp = aggts(p, t)
   compcstemp = compcs(p, t)
   compcompttemp = compcomp(p, t)
   compmstemp = compms(p, t)
   comptstemp = compts(p, t)
'Second compute minimum values
   IF aggcstemp < aggcsmin(p, t) THEN aggcsmin(p, t) = aggcstemp
   IF aggcomptemp < aggcompmin(p, t) THEN aggcompmin(p, t) =
aggcomptemp
   IF aggmstemp < aggmsmin(p, t) THEN aggmsmin(p, t) = aggmstemp
   IF aggtstemp < aggtsmin(p, t) THEN aggtsmin(p, t) = aggtstemp
   IF compcstemp < compcsmin(p, t) THEN compcsmin(p, t) =
compcstemp
   IF compcompttemp < compcompmin(p, t) THEN compcompmin(p, t) =
compcompttemp
   IF compmstemp < compmsmin(p, t) THEN compmsmin(p, t) =
compmstemp
   IF comptstemp < comptsmin(p, t) THEN comptsmin(p, t) =
comptstemp
'Third compute maximum values
   IF aggcstemp > aggcsmax(p, t) THEN aggcsmax(p, t) = aggcstemp
   IF aggcomptemp > aggcompmax(p, t) THEN aggcompmax(p, t) =
aggcomptemp
   IF aggmstemp > aggmsmax(p, t) THEN aggmsmax(p, t) = aggmstemp
   IF aggtstemp > aggtsmax(p, t) THEN aggtsmax(p, t) = aggtstemp
   IF compcstemp > compcsmax(p, t) THEN compcsmax(p, t) =
compcstemp
   IF compcompttemp > compcompmax(p, t) THEN compcompmax(p, t) =
compcompttemp
   IF compmstemp > compmsmax(p, t) THEN compmsmax(p, t) =
compmstemp
   IF comptstemp > comptsmax(p, t) THEN comptsmax(p, t) =
comptstemp
'Fourth compute average values
```

```
comptsavg(16, 3) 'Averages
DIM aggcswtd(16, 3), aggcompwtd(16, 3), aggmswtd(16, 3),
aggtswtd(16, 3) 'Weighted Averages
DIM compcswtd(16, 3), compcompwtd(16, 3), compmswtd(16, 3),
comptswtd(16, 3) 'Weighted Averages

'CONTROL MODULE
CLS
timex = TIMER
   ystart = 1995 'Start year for antitrust analysis.
   yend = 2025 'End year for antitrust analysis.
   cstart = 2002 'Year to start conduct remedies.
   sstart = 2005 'Year to start structural remedies.
GOSUB PVSETUP:
   revstream = 1 '****User chooses revenue stream = 1,2,3,4.
GOSUB READREVENUES: 'Read in Microsoft's revenues by year.
GOSUB READZCOUNT:
'This control module calls the main module 75 times.
FOR cratio = 1 TO 5
   FOR speed = 1 TO 3
      FOR port = 0 TO 4
         GOSUB MAINMODULE:
      NEXT port
   NEXT speed
NEXT cratio
GOSUB PRINTSTATS: 'Print Macro Statistics
PRINT TIMER - timex
END


MAINMODULE:
GOSUB FILENAMES1: 'Assign file name to input & output files.
GOSUB TRANSREAD: 'Read in transitions data.

FOR length = 1 TO 3
   'Tyears = Number of years between transitions.
   '         This program chooses Tyears=3, 5, or 8.
   IF length = 1 THEN tyears = 3
   IF length = 2 THEN tyears = 5
   IF length = 3 THEN tyears = 8
   GOSUB NOREMEDY: 'Compute outcomes for unlawful monopoly path.
   GOSUB LAWFUL: 'Compute outcomes for lawful competitive path.
   GOSUB CONDUCT: 'Compute outcomes for conduct remedy path.
   GOSUB STRUCTURAL: 'Compute outcomes for structural remedy
```

```
paths.
   GOSUB PRINTYEARS: 'Compute and print year data into files.
   GOSUB AGGREGATE: 'Aggregate years for each path.
   GOSUB COMPARE: 'Compare aggregates between paths.
   GOSUB MACROSTATS: 'Compute averages, weighted averages,
minimums, maximums.
   GOSUB PRINTAGGCOMP: 'Print individual aggregates and
comparisons.
   GOSUB PRINTAGGSHORT: 'Print one-page individual aggregates and
comparisons.
'   GOSUB PRINTAGGSUMM: 'Print summary of all aggregates &
comparisons
NEXT length
RETURN
'***** END OF MAIN MODULE *****

PVSETUP:
'Submodule to open file and set initial values
'   for MACROSTATS submodule.
pvfile$ = "c:\basic\ms_sim\PointVal.csv" 'Input Point Values for
weighted averages.
OPEN pvfile$ FOR INPUT AS #21
LINE INPUT #21, temp$
pvtotal = 0
avgtotal = 0
FOR p = 0 TO 14
FOR t = 1 TO 3
'Initialize minimum values at high number.
   aggcsmin(p, t) = 999999999999#
   aggcompmin(p, t) = 999999999999#
   aggmsmin(p, t) = 999999999999#
   aggtsmin(p, t) = 999999999999#
   compcsmin(p, t) = 999999999999#
   compcompmin(p, t) = 999999999999#
   compmsmin(p, t) = 999999999999#
   comptsmin(p, t) = 999999999999#
'Initialize maximum values at low number.
   aggcsmax(p, t) = -999999999999#
   aggcompmax(p, t) = -999999999999#
   aggmsmax(p, t) = -999999999999#
   aggtsmax(p, t) = -999999999999#
   compcsmax(p, t) = -999999999999#
   compcompmax(p, t) = -999999999999#
   compmsmax(p, t) = -999999999999#
```

```
   comptsmax(p, t) = -999999999999#
'Initilize average values at zero.
   aggcsavg(p, t) = 0
   aggcompavg(p, t) = 0
   aggmsavg(p, t) = 0
   aggtsavg(p, t) = 0
   compcsavg(p, t) = 0
   compcompavg(p, t) = 0
   compmsavg(p, t) = 0
   comptsavg(p, t) = 0
'Initialize weighted average values at zero.
   aggcswtd(p, t) = 0
   aggcompwtd(p, t) = 0
   aggmswtd(p, t) = 0
   aggtswtd(p, t) = 0
   compcswtd(p, t) = 0
   compcompwtd(p, t) = 0
   compmswtd(p, t) = 0
   comptswtd(p, t) = 0
NEXT t
NEXT p
RETURN
'***** END OF PVsetup SUBMODULE *****


READREVENUES:
'Read in Microsoft's revenues by year.
'   Revenues should only pertain to the monopoly
'   portions of Microsoft's revenues.
'   Revenues should be converted to real dollars
'   (relative to general prices) prior to input.
'   Future revenues are projections, under the
'   assumption that Microsoft remains a monopoly.
rev$ = "c:\basic\ms_sim\ms_rev.csv" 'Input revenues data.
DIM rev(30), discount(30)
'First read in revenue data.
OPEN rev$ FOR INPUT AS #1
FOR n = 1 TO 5
   LINE INPUT #1, temp$
NEXT n
FOR year = ystart TO yend
   ysub = year - ystart
   INPUT #1, year2, rev1, rev2, rev3, rev4
   IF year2 <> year THEN PRINT "Year mismatch for revenue data:";
```

```
year; year2
   IF revstream = 1 THEN rev(ysub) = rev1
   IF revstream = 2 THEN rev(ysub) = rev2
   IF revstream = 3 THEN rev(ysub) = rev3
   IF revstream = 4 THEN rev(ysub) = rev4
NEXT year
CLOSE #1
'Second compute adjustments to revenue data for computing
aggregates.
'   adjustment uses 7% per annum real interest/discount rate,
'   adjusted to 2002 (cstart) dollars.
adjust = 1
   ysub = (cstart - 1) - ystart
   discount(ysub) = adjust
FOR year = cstart - 2 TO ystart STEP -1
   ysub = year - ystart
   adjust = adjust * 1.07
   discount(ysub) = adjust
NEXT year
adjust = 1
FOR year = cstart TO yend
   ysub = year - ystart
   adjust = adjust / 1.07
   discount(ysub) = adjust
NEXT year
'Third (optional) output adjusted revenue data.
rev2$ = "c:\basic\ms_sim\out\disc_rev.txt"'Output adjusted
revenues data.
OPEN rev2$ FOR OUTPUT AS #1
FOR year = ystart TO yend
   ysub = year - ystart
   PRINT #1, USING "####"; year;
   PRINT #1, USING "#####.######"; rev(ysub);
   PRINT #1, USING "#####.######"; discount(ysub);
   PRINT #1, USING "#####.######"; rev(ysub) * discount(ysub)
NEXT year
CLOSE #1
RETURN
'***** END OF ReadRevenues SUBMODULE *****


READZCOUNT:
'Submodule to read in the relationship between
'   zcount and z from previous program, "MS5TranR.bas".
```

```
DIM zpath(14)
zcount$ = "c:\basic\ms_sim\out\zcount.txt" 'Input zcount data.
OPEN zcount$ FOR INPUT AS #1
   LINE INPUT #1, temp$
FOR zcount = 0 TO 9
   INPUT #1, zcount2, zpath(zcount + 4)
   IF zcount2 <> zcount THEN PRINT "Zcount mismatch"; zcount;
zcount2
NEXT zcount
CLOSE #1
RETURN
'***** END OF ReadZcount SUBMODULE *****


FILENAMES1:
tran1$ = "c:\basic\ms_sim\out\tran1000.txt" 'Input 1-firm
transition summary
tran2$ = "c:\basic\ms_sim\out\tran2000.txt" 'Input 2-firm
transition summary
tran3$ = "c:\basic\ms_sim\out\tran3000.txt" 'Input 3-firm
transition summary
tran4$ = "c:\basic\ms_sim\out\tran4000.txt" 'Input 4-firm
transition summary
tran5$ = "c:\basic\ms_sim\out\tran5000.txt" 'Input 5-firm
transition summary
trpm0$ = "c:\basic\ms_sim\out\trpm0000.txt" 'Input z=0.0 RPM
transition summary
trpm1$ = "c:\basic\ms_sim\out\trpm1000.txt" 'Input z=0.1 RPM
transition summary
trpm2$ = "c:\basic\ms_sim\out\trpm2000.txt" 'Input z=0.2 RPM
transition summary
trpm3$ = "c:\basic\ms_sim\out\trpm3000.txt" 'Input z=0.3 RPM
transition summary
trpm4$ = "c:\basic\ms_sim\out\trpm4000.txt" 'Input z=0.4 RPM
transition summary
trpm5$ = "c:\basic\ms_sim\out\trpm5000.txt" 'Input z=0.5 RPM
transition summary
trpm6$ = "c:\basic\ms_sim\out\trpm6000.txt" 'Input z=0.6 RPM
transition summary
trpm7$ = "c:\basic\ms_sim\out\trpm7000.txt" 'Input z=0.7 RPM
transition summary
trpm8$ = "c:\basic\ms_sim\out\trpm8000.txt" 'Input z=0.8 RPM
transition summary
trpm9$ = "c:\basic\ms_sim\out\trpm9000.txt" 'Input z=0.9 RPM
```

```
transition summary
year3$ = "c:\basic\ms_sim\out\year0003.txt" 'Output 3-year
factors by year
year5$ = "c:\basic\ms_sim\out\year0005.txt" 'Output 5-year
factors by year
year8$ = "c:\basic\ms_sim\out\year0008.txt" 'Output 8-year
factors by year
aggr3$ = "c:\basic\ms_sim\out\aggr0003.txt" 'Output aggregate 3-
year factors
aggr5$ = "c:\basic\ms_sim\out\aggr0005.txt" 'Output aggregate 5-
year factors
aggr8$ = "c:\basic\ms_sim\out\aggr0008.txt" 'Output aggregate 8-
year factors
aggc3$ = "c:\basic\ms_sim\out\aggc0003.txt" 'One-page aggregate
3-year factors
aggc5$ = "c:\basic\ms_sim\out\aggc0005.txt" 'One-page aggregate
5-year factors
aggc8$ = "c:\basic\ms_sim\out\aggc0008.txt" 'One-page aggregate
8-year factors
IF cratio = 1 THEN crt$ = "1"
IF cratio = 2 THEN crt$ = "2"
IF cratio = 3 THEN crt$ = "3"
IF cratio = 4 THEN crt$ = "4"
IF cratio = 5 THEN crt$ = "5"
IF speed = 1 THEN sp$ = "1"
IF speed = 2 THEN sp$ = "2"
IF speed = 3 THEN sp$ = "3"
IF port = 0 THEN prt$ = "0"
IF port = 1 THEN prt$ = "1"
IF port = 2 THEN prt$ = "2"
IF port = 3 THEN prt$ = "3"
IF port = 4 THEN prt$ = "4"
replace$ = crt$ + sp$ + prt$
MID$(tran1$, 26, 3) = replace$
MID$(tran2$, 26, 3) = replace$
MID$(tran3$, 26, 3) = replace$
MID$(tran4$, 26, 3) = replace$
MID$(tran5$, 26, 3) = replace$
MID$(trpm0$, 26, 3) = replace$
MID$(trpm1$, 26, 3) = replace$
MID$(trpm2$, 26, 3) = replace$
MID$(trpm3$, 26, 3) = replace$
MID$(trpm4$, 26, 3) = replace$
MID$(trpm5$, 26, 3) = replace$
```

```
MID$(trpm6$, 26, 3) = replace$
MID$(trpm7$, 26, 3) = replace$
MID$(trpm8$, 26, 3) = replace$
MID$(trpm9$, 26, 3) = replace$
MID$(year3$, 25, 3) = replace$
MID$(year5$, 25, 3) = replace$
MID$(year8$, 25, 3) = replace$
MID$(aggr3$, 25, 3) = replace$
MID$(aggr5$, 25, 3) = replace$
MID$(aggr8$, 25, 3) = replace$
MID$(aggc3$, 25, 3) = replace$
MID$(aggc5$, 25, 3) = replace$
MID$(aggc8$, 25, 3) = replace$
PRINT replace$; " ";
RETURN
'***** END OF FileNames1 SUBMODULE *****


TRANSREAD:
'Submodule to read in transitions data for time paths:
'   1) Microsoft starts as monopoly.
'   2) Microsoft starts as two APM firms.
'   3) Microsoft starts as three APM firms.
'   4-13) Microsoft starts as two RPM firms (z varies).
FOR p = 1 TO 13
IF p = 1 THEN filein$ = tran1$
IF p = 2 THEN filein$ = tran2$
IF p = 3 THEN filein$ = tran3$
IF p = 4 THEN filein$ = trpm0$
IF p = 5 THEN filein$ = trpm1$
IF p = 6 THEN filein$ = trpm2$
IF p = 7 THEN filein$ = trpm3$
IF p = 8 THEN filein$ = trpm4$
IF p = 9 THEN filein$ = trpm5$
IF p = 10 THEN filein$ = trpm6$
IF p = 11 THEN filein$ = trpm7$
IF p = 12 THEN filein$ = trpm8$
IF p = 13 THEN filein$ = trpm9$
OPEN filein$ FOR INPUT AS #30
LINE INPUT #30, temp$
DIM cstot(15, 11), pimstot(15, 11), picomptot(15, 11), herf(15,
11)
FOR iter = 0 TO 10
   INPUT #30, iter2, cstot(p, iter), pimstot(p, iter),
```

```
picomptot(p, iter)
   INPUT #30, mktshare1, mktshare2, herf(p, iter), firmnum1
     IF iter <> iter2 THEN PRINT "Iteration mismatch #", p, iter,
iter1
NEXT iter
CLOSE #30
cstot(p, 11) = 0
pimstot(p, 11) = 0
picomptot(p, 11) = 0
herf(p, 11) = 0
NEXT p
RETURN
'***** END OF TransRead SUBMODULE *****


NOREMEDY:
'Submodule to compute outcomes for unlawful monopoly path,
'   where Microsoft begins as monopoly in 1995,
'   and continues as a monopoly through 2025.  (Path p=0)
DIM csy(30, 14), pimsy(30, 14), picompy(30, 14), hhi(30, 14)
FOR year = ystart TO yend
   ysub = year - ystart
   tsub = 0
   csy(ysub, 0) = cstot(1, tsub)
   pimsy(ysub, 0) = pimstot(1, tsub)
   picompy(ysub, 0) = picomptot(1, tsub)
   hhi(ysub, 0) = herf(1, tsub)
NEXT year
RETURN
'***** END OF NoRemedy SUBMODULE *****


LAWFUL:
'Submodule to compute outcomes for lawful competitive path,
'   where Microsoft begins as monopoly in 1995,
'   but competitive conditions exist whereby
'   competitors are free to enter.  (Path p=14)
plaw = 14
FOR year = ystart TO yend
   ysub = year - ystart
   tsub = ysub / tyears
   tsub1 = INT(tsub)
   tsub2 = tsub1 + 1
   tfrac1 = tsub - tsub1
```

```
   tfrac2 = tsub2 - tsub
   csy(ysub, plaw) = cstot(1, tsub1) * tfrac2 + cstot(1, tsub2) *
tfrac1
   pimsy(ysub, plaw) = pimstot(1, tsub1) * tfrac2 + pimstot(1,
tsub2) * tfrac1
   picompy(ysub, plaw) = picomptot(1, tsub1) * tfrac2 +
picomptot(1, tsub2) * tfrac1
   hhi(ysub, plaw) = herf(1, tsub1) * tfrac2 + herf(1, tsub2) *
tfrac1
NEXT year
RETURN
'***** END OF Lawful SUBMODULE *****


CONDUCT:
'Submodule to compute outcomes for conduct remedy path,
'   where Microsoft exists as a monopoly in 1995-2001,
'   but competitive conditions begin in 2002 (cstart)
'   whereby competitors are free to enter.  (Path p=1)
FOR year = ystart TO cstart - 1
   ysub = year - ystart
   tsub = 0
   csy(ysub, 1) = cstot(1, tsub)
   pimsy(ysub, 1) = pimstot(1, tsub)
   picompy(ysub, 1) = picomptot(1, tsub)
   hhi(ysub, 1) = herf(1, tsub)
NEXT year
cstart1 = cstart - 1
FOR year = cstart TO yend
   ysub = year - ystart
   tsub = (year - cstart1) / tyears
   tsub1 = INT(tsub)
   tsub2 = tsub1 + 1
   tfrac1 = tsub - tsub1
   tfrac2 = tsub2 - tsub
   csy(ysub, 1) = cstot(1, tsub1) * tfrac2 + cstot(1, tsub2) *
tfrac1
   pimsy(ysub, 1) = pimstot(1, tsub1) * tfrac2 + pimstot(1,
tsub2) * tfrac1
   picompy(ysub, 1) = picomptot(1, tsub1) * tfrac2 + picomptot(1,
tsub2) * tfrac1
   hhi(ysub, 1) = herf(1, tsub1) * tfrac2 + herf(1, tsub2) *
tfrac1
NEXT year
```

```
RETURN
'***** END OF Conduct SUBMODULE *****


STRUCTURAL:
'Submodule to compute outcomes for structural remedy paths,
'   where Microsoft exists as a monopoly in 1995-2004,
'   but Microsoft is divided into 2 or 3 firms in 2005
'   and competitive conditions exist thereafter.
'   Path p=2, Microsoft divided into 2 APM firms.
'   Path p=3, Microsoft divided into 3 APM firms.
'   Paths p=4 thru p=13, Microsoft divided into 2 RPM firms,
'   where z is allowed to vary.
FOR p = 2 TO 13
FOR year = ystart TO sstart - 1
   ysub = year - ystart
   tsub = 0
   csy(ysub, p) = cstot(1, tsub)
   pimsy(ysub, p) = pimstot(1, tsub)
   picompy(ysub, p) = picomptot(1, tsub)
   hhi(ysub, p) = herf(1, tsub)
NEXT year
FOR year = sstart TO yend
   ysub = year - ystart
   tsub = (year - sstart) / tyears
   tsub1 = INT(tsub)
   tsub2 = tsub1 + 1
   tfrac1 = tsub - tsub1
   tfrac2 = tsub2 - tsub
   csy(ysub, p) = cstot(p, tsub1) * tfrac2 + cstot(p, tsub2) *
tfrac1
   pimsy(ysub, p) = pimstot(p, tsub1) * tfrac2 + pimstot(p,
tsub2) * tfrac1
   picompy(ysub, p) = picomptot(p, tsub1) * tfrac2 + picomptot(p,
tsub2) * tfrac1
   hhi(ysub, p) = herf(p, tsub1) * tfrac2 + herf(p, tsub2) *
tfrac1
NEXT year
NEXT p
RETURN
'***** END OF Structural SUBMODULE *****


PRINTYEARS:
```

```
'Submodule to compute and print the data by year into files.
'   Computed data consists of consumer surplus (cs), profits for
'   Microsoft (pims), profits for competitors (picomp),
'   and Herfindahl-Hershman Index (HHI).
'   Data is computed for several time paths:
'   No remedy (continued monopoly) path=0;
'   Perfect conduct remedy starting in 2002 (path=1);
'   Structural remedies starting in 2005 (paths=2-13);
'   Lawful path (competitive behavior) since 1995 (path=14).
'First adjust year factors by multiplying with revenue data.
'   The computed year data is expressed in real terms,
'   but is not adjusted for 7% interest/discount rate.
FOR year = ystart TO yend
   ysub = year - ystart
   FOR path = 0 TO 14
      csy(ysub, path) = csy(ysub, path) * rev(ysub)
      pimsy(ysub, path) = pimsy(ysub, path) * rev(ysub)
      picompy(ysub, path) = picompy(ysub, path) * rev(ysub)
   NEXT path
NEXT year
'Second (optional) print the year data into files.
IF tyears = 3 THEN OPEN year3$ FOR OUTPUT AS #6
IF tyears = 5 THEN OPEN year5$ FOR OUTPUT AS #6
IF tyears = 8 THEN OPEN year8$ FOR OUTPUT AS #6
FOR year = ystart TO yend
   ysub = year - ystart
   PRINT #6, USING "####"; year;
   FOR path = 0 TO 14
      PRINT #6, USING "#####.#########"; csy(ysub, path);
   NEXT path
   PRINT #6,
NEXT year
   PRINT #6,
FOR year = ystart TO yend
   ysub = year - ystart
   PRINT #6, USING "####"; year;
   FOR path = 0 TO 14
      PRINT #6, USING "#####.#########"; pimsy(ysub, path);
   NEXT path
   PRINT #6,
NEXT year
   PRINT #6,
FOR year = ystart TO yend
   ysub = year - ystart
```

```
    PRINT #6, USING "####"; year;
    FOR path = 0 TO 14
       PRINT #6, USING "#####.#########"; picompy(ysub, path);
    NEXT path
    PRINT #6,
NEXT year
    PRINT #6,
FOR year = ystart TO yend
    ysub = year - ystart
    PRINT #6, USING "####"; year;
    FOR path = 0 TO 14
       tstot = csy(ysub, path) + pimsy(ysub, path) + picompy(ysub,
path)
       PRINT #6, USING "#####.#########"; tstot;
    NEXT path
    PRINT #6,
NEXT year
    PRINT #6,
FOR year = ystart TO yend
    ysub = year - ystart
    PRINT #6, USING "####"; year;
    FOR path = 0 TO 14
       PRINT #6, USING "######.########"; hhi(ysub, path);
    NEXT path
    PRINT #6,
NEXT year
CLOSE #6
RETURN
'***** END OF PrintYears SUBMODULE *****


AGGREGATE:
'Submodule to aggregate years for each path.
'  Agg..(p,t) is the aggregate for path p, time period t.
'    p=0, no remedy path;
'    p=1, 1-firm conduct remedy path;
'    p=2, 2-firm structural remedy path.
'    p=3, 3-firm structural remedy path.
'    p=4 to p=13, 2-firm RPM structural remedy path.
'    p=14, lawful path;
'    t=1, 1995-2001; t=2, 2002-2025; t=3, 1995-2025.
'  AggCS = Aggregate for consumer surplus.
'  AggComp = Aggregate for non-Microsoft profits.
'  AggMS = Aggregate for Microsoft's profits.
```

```
' AggTS = Aggregate for total surplus.
'First adjust the year data for 7% discount/interest rate.
FOR year = ystart TO yend
   ysub = year - ystart
   FOR path = 0 TO 14
      csy(ysub, path) = csy(ysub, path) * discount(ysub)
      pimsy(ysub, path) = pimsy(ysub, path) * discount(ysub)
      picompy(ysub, path) = picompy(ysub, path) * discount(ysub)
   NEXT path
NEXT year
'Second aggregate the data for the three time periods.
FOR path = 0 TO 14
      cstemp = 0
      picomptemp = 0
      mstemp = 0
   FOR year = ystart TO cstart - 1
      ysub = year - ystart
      cstemp = cstemp + csy(ysub, path)
      picomptemp = picomptemp + picompy(ysub, path)
      mstemp = mstemp + pimsy(ysub, path)
   NEXT year
      aggcs(path, 1) = cstemp
      aggcomp(path, 1) = picomptemp
      aggms(path, 1) = mstemp
      aggts(path, 1) = cstemp + picomptemp + mstemp
      cstemp = 0
      picomptemp = 0
      mstemp = 0
   FOR year = cstart TO yend
      ysub = year - ystart
      cstemp = cstemp + csy(ysub, path)
      picomptemp = picomptemp + picompy(ysub, path)
      mstemp = mstemp + pimsy(ysub, path)
   NEXT year
      aggcs(path, 2) = cstemp
      aggcomp(path, 2) = picomptemp
      aggms(path, 2) = mstemp
      aggts(path, 2) = cstemp + picomptemp + mstemp
      aggcs(path, 3) = aggcs(path, 1) + aggcs(path, 2)
      aggcomp(path, 3) = aggcomp(path, 1) + aggcomp(path, 2)
      aggms(path, 3) = aggms(path, 1) + aggms(path, 2)
      aggts(path, 3) = aggts(path, 1) + aggts(path, 2)
NEXT path
RETURN
```

```
'***** END OF Aggregate SUBMODULE *****


COMPARE:
'Submodule to compare aggregates between different time paths.
'  Comp..(c,t) is comparison c for time period t.
'    c=0, no remedy path minus lawful competitive path.
'    c=1, 1-firm conduct remedy path minus lawful competitive
path.
'    c=2, 2-firm structural remedy path minus lawful competitive
path.
'    c=3, 3-firm structural remedy path minus lawful competitive
path.
'    c=4, 4-firm structural remedy path minus lawful competitive
path.
'    c=5, 5-firm structural remedy path minus lawful competitive
path.
'    t=1, 1995-2001; t=2, 2002-2025; t=3, 1995-2025;
'  CompCS = Comparison for consumer surplus.
'  CompComp = Comparison non-Microsoft profits.
'  CompMS = Comparison for Microsoft's profits.
'  CompTS = Comparison for total surplus.
FOR c = 0 TO 13
   FOR t = 1 TO 3
      compcs(c, t) = aggcs(c, t) - aggcs(plaw, t)
      compcomp(c, t) = aggcomp(c, t) - aggcomp(plaw, t)
      compms(c, t) = aggms(c, t) - aggms(plaw, t)
      compts(c, t) = aggts(c, t) - aggts(plaw, t)
   NEXT t
NEXT c
RETURN
'***** END OF Compare SUBMODULE *****


MACROSTATS:
'Submodule to compute averages, weighted averages, minimums,
maximums.
INPUT #21, cratio2, speed2, port2, tyears2, pvtemp
   IF cratio <> cratio2 THEN PRINT "Cost-ratio mismatch:";
cratio; cratio2
   IF speed <> speed2 THEN PRINT "Speed mismatch:"; speed; speed2
   IF tyears <> tyears2 THEN PRINT "Tyears mismatch:"; tyears;
tyears2
   IF port <> port2 THEN PRINT "Portion mismatch:"; port; port2
```

```
IF port >= 0 THEN 'Always true; change to restrict statistics
gathering.
   pvtotal = pvtotal + pvtemp
   avgtotal = avgtotal + 1
FOR p = 0 TO 14
FOR t = 1 TO 3
'First put numbers into temporary variable slots.
   aggcstemp = aggcs(p, t)
   aggcomptemp = aggcomp(p, t)
   aggmstemp = aggms(p, t)
   aggtstemp = aggts(p, t)
   compcstemp = compcs(p, t)
   compcompttemp = compcomp(p, t)
   compmstemp = compms(p, t)
   comptstemp = compts(p, t)
'Second compute minimum values
   IF aggcstemp < aggcsmin(p, t) THEN aggcsmin(p, t) = aggcstemp
   IF aggcomptemp < aggcompmin(p, t) THEN aggcompmin(p, t) =
aggcomptemp
   IF aggmstemp < aggmsmin(p, t) THEN aggmsmin(p, t) = aggmstemp
   IF aggtstemp < aggtsmin(p, t) THEN aggtsmin(p, t) = aggtstemp
   IF compcstemp < compcsmin(p, t) THEN compcsmin(p, t) =
compcstemp
   IF compcompttemp < compcompmin(p, t) THEN compcompmin(p, t) =
compcompttemp
   IF compmstemp < compmsmin(p, t) THEN compmsmin(p, t) =
compmstemp
   IF comptstemp < comptsmin(p, t) THEN comptsmin(p, t) =
comptstemp
'Third compute maximum values
   IF aggcstemp > aggcsmax(p, t) THEN aggcsmax(p, t) = aggcstemp
   IF aggcomptemp > aggcompmax(p, t) THEN aggcompmax(p, t) =
aggcomptemp
   IF aggmstemp > aggmsmax(p, t) THEN aggmsmax(p, t) = aggmstemp
   IF aggtstemp > aggtsmax(p, t) THEN aggtsmax(p, t) = aggtstemp
   IF compcstemp > compcsmax(p, t) THEN compcsmax(p, t) =
compcstemp
   IF compcompttemp > compcompmax(p, t) THEN compcompmax(p, t) =
compcompttemp
   IF compmstemp > compmsmax(p, t) THEN compmsmax(p, t) =
compmstemp
   IF comptstemp > comptsmax(p, t) THEN comptsmax(p, t) =
comptstemp
'Fourth compute average values
```

```
    aggcsavg(p, t) = aggcsavg(p, t) + aggcstemp
    aggcompavg(p, t) = aggcompavg(p, t) + aggcomptemp
    aggmsavg(p, t) = aggmsavg(p, t) + aggmstemp
    aggtsavg(p, t) = aggtsavg(p, t) + aggtstemp
    compcsavg(p, t) = compcsavg(p, t) + compcstemp
    compcompavg(p, t) = compcompavg(p, t) + compcompttemp
    compmsavg(p, t) = compmsavg(p, t) + compmstemp
    comptsavg(p, t) = comptsavg(p, t) + comptstemp
 'Fifth compute weighted average values
    aggcswtd(p, t) = aggcswtd(p, t) + aggcstemp * pvtemp
    aggcompwtd(p, t) = aggcompwtd(p, t) + aggcomptemp * pvtemp
    aggmswtd(p, t) = aggmswtd(p, t) + aggmstemp * pvtemp
    aggtswtd(p, t) = aggtswtd(p, t) + aggtstemp * pvtemp
    compcswtd(p, t) = compcswtd(p, t) + compcstemp * pvtemp
    compcompwtd(p, t) = compcompwtd(p, t) + compcompttemp * pvtemp
    compmswtd(p, t) = compmswtd(p, t) + compmstemp * pvtemp
    comptswtd(p, t) = comptswtd(p, t) + comptstemp * pvtemp
 NEXT t
 NEXT p
 END IF
 RETURN
 '***** END OF MacroStats SUBMODULE *****


 PRINTAGGCOMP:
 'Submodule to print aggregates and comparisons.
 IF tyears = 3 THEN OPEN aggr3$ FOR OUTPUT AS #7
 IF tyears = 5 THEN OPEN aggr5$ FOR OUTPUT AS #7
 IF tyears = 8 THEN OPEN aggr8$ FOR OUTPUT AS #7
 'First print aggregates.
 FOR path = 0 TO 14
    IF path = 0 THEN PRINT #7, "              Aggregates   for
 No Remedy     Path:"
    IF path = 1 THEN PRINT #7, "              Aggregates   for
 Conduct       Remedy:"
    IF path = 2 THEN PRINT #7, "              Aggregates   for
 APM,2-firms   Remedy:"
    IF path = 3 THEN PRINT #7, "              Aggregates   for
 APM,3-firms   Remedy:"
    IF path >= 4 AND path <= 13 THEN
        PRINT #7, "         Aggregates    for          RPM,z=";
        PRINT #7, USING "#.###"; zpath(path);
        PRINT #7, "  Remedy:"
        END IF
```

```
   IF path = 14 THEN PRINT #7, "         Aggregates    for
Lawful        Path:"
   PRINT #7, " Time    "; "       CS      "; "   nonMSpi    "; "
MSpi    "; "       TS      "
   FOR t = 1 TO 3
      IF t = 1 THEN PRINT #7, "Past:    ";
      IF t = 2 THEN PRINT #7, "Future: ";
      IF t = 3 THEN PRINT #7, "Total:   ";
      PRINT #7, USING "#####.#######"; aggcs(path, t);
aggcomp(path, t); aggms(path, t); aggts(path, t)
   NEXT t
   PRINT #7,
NEXT path
'Second print comparisons.
FOR c = 0 TO 13
   IF c = 0 THEN PRINT #7, "          Comparing    No Remedy
minus         LawfulPath:"
   IF c = 1 THEN PRINT #7, "          Comparing    Conduct
minus         LawfulPath:"
   IF c = 2 THEN PRINT #7, "          Comparing    APM,2-firms
minus         LawfulPath:"
   IF c = 3 THEN PRINT #7, "          Comparing    APM,3-firms
minus         LawfulPath:"
   IF c >= 4 AND c <= 13 THEN
      PRINT #7, "          Comparing    RPM,z=";
      PRINT #7, USING "#.###"; zpath(c);
      PRINT #7, "  minus         LawfulPath:"
      END IF
   PRINT #7, " Time    "; "       CS      "; "   nonMSpi    "; "
MSpi    "; "       TS      "
   FOR t = 1 TO 3
      IF t = 1 THEN PRINT #7, "Past:    ";
      IF t = 2 THEN PRINT #7, "Future: ";
      IF t = 3 THEN PRINT #7, "Total:   ";
      PRINT #7, USING "#####.#######"; compcs(c, t); compcomp(c,
t); compms(c, t); compts(c, t)
   NEXT t
   PRINT #7,
NEXT c
CLOSE #7
RETURN
'***** END OF PrintAggComp SUBMODULE *****
```

```
PRINTAGGSHORT:
'Submodule to print one-page summary of
'   aggregates and comparisons.
IF tyears = 3 THEN OPEN aggc3$ FOR OUTPUT AS #7
IF tyears = 5 THEN OPEN aggc5$ FOR OUTPUT AS #7
IF tyears = 8 THEN OPEN aggc8$ FOR OUTPUT AS #7
'First print totals for alternative remedies.
   PRINT #7, "                  Total       Aggregates   Alternative
Remedies:"
   PRINT #7, "  Remedy      "; "      CS      "; "    nonMSpi   "; "
   MSpi    "; "       TS      "
FOR path = 0 TO 14
   IF path = 0 THEN PRINT #7, "No-Remedy:    ";
   IF path = 1 THEN GOSUB AGGSUB:
   IF path = 1 THEN path = 2
   IF path = 2 THEN PRINT #7, "APM,2-firms: ";
   IF path = 3 THEN PRINT #7, "APM,3-firms: ";
   IF path >= 4 AND path <= 13 THEN
      PRINT #7, "RPM,z=";
      PRINT #7, USING "#.###"; zpath(path);
      PRINT #7, ": ";
      END IF
   IF path = 14 THEN PRINT #7, "Lawful Path: ";
   PRINT #7, USING "#####.#######"; aggcs(path, 3); aggcomp(path,
3); aggms(path, 3); aggts(path, 3)
NEXT path
PRINT #7,
'Second print comparisons.
   PRINT #7, "                  Comparing    Remedies     minus
Lawful Path:"
   PRINT #7, "  Remedy      "; "      CS      "; "    nonMSpi   "; "
   MSpi    "; "       TS      "
FOR c = 0 TO 13
   IF c = 0 THEN PRINT #7, "No-Remedy:    ";
   IF c = 1 THEN GOSUB COMPSUB:
   IF c = 1 THEN c = 2
   IF c = 2 THEN PRINT #7, "APM,2-firms: ";
   IF c = 3 THEN PRINT #7, "APM,3-firms: ";
   IF c >= 4 AND c <= 13 THEN
      PRINT #7, "RPM,z=";
      PRINT #7, USING "#.###"; zpath(c);
      PRINT #7, ": ";
      END IF
   PRINT #7, USING "#####.#######"; compcs(c, 3); compcomp(c, 3);
```

```
compms(c, 3); compts(c, 3)
NEXT c
CLOSE #7
RETURN
'***** END OF PrintAggShort SUBMODULE *****


AGGSUB:
'Submodule of PRINTAGGSHORT Submodule,
'   to print out variations on conduct remedy.
   temp0cs = aggcs(0, 3)
   temp0comp = aggcomp(0, 3)
   temp0ms = aggms(0, 3)
   temp0ts = aggts(0, 3)
   temp1cs = aggcs(1, 3)
   temp1comp = aggcomp(1, 3)
   temp1ms = aggms(1, 3)
   temp1ts = aggts(1, 3)
   temp2cs = aggcs(2, 3)
   temp2comp = aggcomp(2, 3)
   temp2ms = aggms(2, 3)
   temp2ts = aggts(2, 3)
GOSUB PRINTSUB:
RETURN
'***** END OF AggSub SUBMODULE *****


COMPSUB:
'Submodule of PRINTAGGSHORT Submodule,
'   to print out variations on conduct remedy.
   temp0cs = compcs(0, 3)
   temp0comp = compcomp(0, 3)
   temp0ms = compms(0, 3)
   temp0ts = compts(0, 3)
   temp1cs = compcs(1, 3)
   temp1comp = compcomp(1, 3)
   temp1ms = compms(1, 3)
   temp1ts = compts(1, 3)
   temp2cs = compcs(2, 3)
   temp2comp = compcomp(2, 3)
   temp2ms = compms(2, 3)
   temp2ts = compts(2, 3)
GOSUB PRINTSUB:
RETURN
```

```
'***** END OF CompSub SUBMODULE *****


PRINTSUB:
'Submodule of two submodules of the PRINTAGGSHORT
'   submodule, to print variations on conduct remedy.
'Compute and print 20% effective conduct remedy.
   PRINT #7, " 20% Conduct:";
   PRINT #7, USING "#####.#######"; temp0cs * .8 + temp1cs * .2;
   PRINT #7, USING "#####.#######"; temp0comp * .8 + temp1comp *
.2;
   PRINT #7, USING "#####.#######"; temp0ms * .8 + temp1ms * .2;
   PRINT #7, USING "#####.#######"; temp0ts * .8 + temp1ts * .2
'Compute and print 40% effective conduct remedy.
   PRINT #7, " 40% Conduct:";
   PRINT #7, USING "#####.#######"; temp0cs * .6 + temp1cs * .4;
   PRINT #7, USING "#####.#######"; temp0comp * .6 + temp1comp *
.4;
   PRINT #7, USING "#####.#######"; temp0ms * .6 + temp1ms * .4;
   PRINT #7, USING "#####.#######"; temp0ts * .6 + temp1ts * .4
'Compute and print 60% effective conduct remedy.
   PRINT #7, " 60% Conduct:";
   PRINT #7, USING "#####.#######"; temp0cs * .4 + temp1cs * .6;
   PRINT #7, USING "#####.#######"; temp0comp * .4 + temp1comp *
.6;
   PRINT #7, USING "#####.#######"; temp0ms * .4 + temp1ms * .6;
   PRINT #7, USING "#####.#######"; temp0ts * .4 + temp1ts * .6
'Compute and print 80% effective conduct remedy.
   PRINT #7, " 80% Conduct:";
   PRINT #7, USING "#####.#######"; temp0cs * .2 + temp1cs * .8;
   PRINT #7, USING "#####.#######"; temp0comp * .2 + temp1comp *
.8;
   PRINT #7, USING "#####.#######"; temp0ms * .2 + temp1ms * .8;
   PRINT #7, USING "#####.#######"; temp0ts * .2 + temp1ts * .8
'Compute and print 100% effective conduct remedy.
   PRINT #7, "100% Conduct:";
   PRINT #7, USING "#####.#######"; temp1cs;
   PRINT #7, USING "#####.#######"; temp1comp;
   PRINT #7, USING "#####.#######"; temp1ms;
   PRINT #7, USING "#####.#######"; temp1ts
'Compute and print the two-monopolies structural remedy.
   PRINT #7, "2-Monopolies:";
   PRINT #7, USING "#####.#######"; temp2cs / 3 + temp1cs * 2 /
3;
```

```
   PRINT #7, USING "#####.#######"; temp2comp / 3 + templcomp * 2
/ 3;
   PRINT #7, USING "#####.#######"; temp2ms / 3 + templms * 2 /
3;
   PRINT #7, USING "#####.#######"; temp2ts / 3 + templts * 2 / 3
RETURN
'***** END OF PrintSub SUBMODULE *****


PRINTSTATS:
'Submodule to print averages, weighted averages, minimums,
maximums.
CLOSE #21
'First print out minimums.
MID$(aggr8$, 25, 3) = "MIN"
MID$(aggc8$, 25, 3) = "MIN"
FOR p = 0 TO 14
FOR t = 1 TO 3
   aggcs(p, t) = aggcsmin(p, t)
   aggcomp(p, t) = aggcompmin(p, t)
   aggms(p, t) = aggmsmin(p, t)
   aggts(p, t) = aggtsmin(p, t)
   compcs(p, t) = compcsmin(p, t)
   compcomp(p, t) = compcompmin(p, t)
   compms(p, t) = compmsmin(p, t)
   compts(p, t) = comptsmin(p, t)
NEXT t
NEXT p
GOSUB PRINTAGGCOMP:
GOSUB PRINTAGGSHORT:
'Second print out maximums.
MID$(aggr8$, 25, 3) = "MAX"
MID$(aggc8$, 25, 3) = "MAX"
FOR p = 0 TO 14
FOR t = 1 TO 3
   aggcs(p, t) = aggcsmax(p, t)
   aggcomp(p, t) = aggcompmax(p, t)
   aggms(p, t) = aggmsmax(p, t)
   aggts(p, t) = aggtsmax(p, t)
   compcs(p, t) = compcsmax(p, t)
   compcomp(p, t) = compcompmax(p, t)
   compms(p, t) = compmsmax(p, t)
   compts(p, t) = comptsmax(p, t)
NEXT t
```

```
NEXT p
GOSUB PRINTAGGCOMP:
GOSUB PRINTAGGSHORT:
MID$(aggr8$, 25, 3) = "AVG"
MID$(aggc8$, 25, 3) = "AVG"
FOR p = 0 TO 14
FOR t = 1 TO 3
   aggcs(p, t) = aggcsavg(p, t) / avgtotal
   aggcomp(p, t) = aggcompavg(p, t) / avgtotal
   aggms(p, t) = aggmsavg(p, t) / avgtotal
   aggts(p, t) = aggtsavg(p, t) / avgtotal
   compcs(p, t) = compcsavg(p, t) / avgtotal
   compcomp(p, t) = compcompavg(p, t) / avgtotal
   compms(p, t) = compmsavg(p, t) / avgtotal
   compts(p, t) = comptsavg(p, t) / avgtotal
NEXT t
NEXT p
GOSUB PRINTAGGCOMP:
GOSUB PRINTAGGSHORT:
'Fourth print out weighted averages.
MID$(aggr8$, 25, 3) = "WTD"
MID$(aggc8$, 25, 3) = "WTD"
FOR p = 0 TO 14
FOR t = 1 TO 3
   aggcs(p, t) = aggcswtd(p, t) / pvtotal
   aggcomp(p, t) = aggcompwtd(p, t) / pvtotal
   aggms(p, t) = aggmswtd(p, t) / pvtotal
   aggts(p, t) = aggtswtd(p, t) / pvtotal
   compcs(p, t) = compcswtd(p, t) / pvtotal
   compcomp(p, t) = compcompwtd(p, t) / pvtotal
   compms(p, t) = compmswtd(p, t) / pvtotal
   compts(p, t) = comptswtd(p, t) / pvtotal
NEXT t
NEXT p
GOSUB PRINTAGGCOMP:
GOSUB PRINTAGGSHORT:
RETURN
'***** END OF PrintStats SUBMODULE *****

'**********END OF Program "MS6Summ.bas".**********

'**********LAST OF SERIES OF SIX PROGRAMS**********
```

Attachment R.

Contents of File "MS_Rev.csv".
The "MS_Rev.csv" file is a needed input file for the "MS6Summ.bas" computer program.

```
Microsoft Corporation,,,,
"Real Annual Revenue, 2001 dollars (billions).",,,,
,,,,"Platforms,"
Calendar,Desktop,Platforms &,Platforms &,Applications
Year,Platforms,Enterprise,Applications,& Enterprise
1995,3.0035316,4.207892342,6.855180159,8.059540901
1996,3.727131393,5.347349902,8.460443423,10.08066193
1997,5.035883238,7.458570139,11.21720468,13.63989158
1998,6.454595209,9.391381895,14.2045433,17.14132999
1999,7.693463232,12.21087139,17.14963014,21.6670383
2000,8.186611784,13.1922708,17.72984147,22.73550048
2001,7.2043035,11.34811009,16.7864329,20.93023949
2002,9.14247463,14.95519357,19.79287211,25.60559105
2003,10.58883605,17.7114904,22.8211218,29.94377615
2004,12.00140966,20.45382314,25.77105291,34.22346639
2005,13.36456237,23.13621511,28.61300897,38.38466171
2006,14.67011841,25.72877347,31.33233776,42.39099283
2007,15.91566918,28.21539046,33.92583204,46.22555332
2008,17.10294472,30.59079228,36.39838725,49.88623481
2009,18.23641933,32.85759662,38.760186,53.38136329
2010,19.3222093,35.02373565,41.02449912,56.72602548
2011,20.36725465,37.10038404,43.20607024,59.93919963
2012,21.3787453,39.10039929,45.31999424,63.04164823
2013,22.36374177,41.03721389,47.38098491,66.05445702
2014,23.32894261,42.92409405,49.40293308,68.99808453
2015,24.28055711,44.77367791,51.39867211,71.89179291
2016,25.22425072,46.59771715,53.37988484,74.75335127
2017,26.1651381,48.40695999,55.35710303,77.59892492
2018,27.107806,50.21112823,57.33976344,80.44308567
2019,28.05635302,52.01895334,59.33629582,83.29889613
2020,29.01443792,53.83824715,61.35422596,86.17803519
2021,29.98533062,55.6759901,63.40028299,89.09094247
2022,30.97196248,57.5384259,65.48050406,92.04696748
2023,31.97697372,59.43115563,67.60033258,95.05451449
2024,33.002757,61.35922702,69.76470794,98.12117797
2025,34.0514965,63.32721672,71.97814598,101.2538662
2026,35.1252027,65.3393044,74.24481019,104.4589119
2027,36.22574276,67.39933867,76.56857413,107.74217
2028,37.35486715,69.51089506,78.95307569,111.1091036
2029,38.51423258,71.67732649,81.40176416,114.5648581
2030,39.70542191,73.90180714,83.91794075,118.114326
```

Contents of File "PointVal.csv".
The "PointVal.csv" file is a needed input file for the "MS6Summ.bas" computer program.

```
Cost-ratio,Speed,Portion,T-years,Point Values
1,1,0,3,1
1,1,0,5,1
1,1,0,8,1
1,1,1,3,2
1,1,1,5,2
1,1,1,8,2
1,1,2,3,2
1,1,2,5,2
1,1,2,8,2
1,1,3,3,2
1,1,3,5,2
1,1,3,8,2
1,1,4,3,1
1,1,4,5,1
1,1,4,8,1
1,2,0,3,1
1,2,0,5,1
1,2,0,8,1
1,2,1,3,2
1,2,1,5,2
1,2,1,8,2
1,2,2,3,2
1,2,2,5,2
1,2,2,8,2
1,2,3,3,2
1,2,3,5,2
1,2,3,8,2
1,2,4,3,1
1,2,4,5,1
1,2,4,8,1
1,3,0,3,1
1,3,0,5,1
1,3,0,8,1
1,3,1,3,2
1,3,1,5,2
1,3,1,8,2
1,3,2,3,2
1,3,2,5,2
1,3,2,8,2
1,3,3,3,2
1,3,3,5,2
1,3,3,8,2
1,3,4,3,1
1,3,4,5,1
1,3,4,8,1
```

2,1,0,3,1
2,1,0,5,1
2,1,0,8,1
2,1,1,3,2
2,1,1,5,2
2,1,1,8,2
2,1,2,3,2
2,1,2,5,2
2,1,2,8,2
2,1,3,3,2
2,1,3,5,2
2,1,3,8,2
2,1,4,3,1
2,1,4,5,1
2,1,4,8,1
2,2,0,3,1
2,2,0,5,1
2,2,0,8,1
2,2,1,3,2
2,2,1,5,2
2,2,1,8,2
2,2,2,3,2
2,2,2,5,2
2,2,2,8,2
2,2,3,3,2
2,2,3,5,2
2,2,3,8,2
2,2,4,3,1
2,2,4,5,1
2,2,4,8,1
2,3,0,3,1
2,3,0,5,1
2,3,0,8,1
2,3,1,3,2
2,3,1,5,2
2,3,1,8,2
2,3,2,3,2
2,3,2,5,2
2,3,2,8,2
2,3,3,3,2
2,3,3,5,2
2,3,3,8,2
2,3,4,3,1
2,3,4,5,1
2,3,4,8,1

3,1,0,3,2
3,1,0,5,2
3,1,0,8,2
3,1,1,3,4
3,1,1,5,4
3,1,1,8,4
3,1,2,3,4
3,1,2,5,4
3,1,2,8,4
3,1,3,3,4
3,1,3,5,4
3,1,3,8,4
3,1,4,3,2
3,1,4,5,2
3,1,4,8,2
3,2,0,3,2
3,2,0,5,2
3,2,0,8,2
3,2,1,3,4
3,2,1,5,4
3,2,1,8,4
3,2,2,3,4
3,2,2,5,4
3,2,2,8,4
3,2,3,3,4
3,2,3,5,4
3,2,3,8,4
3,2,4,3,2
3,2,4,5,2
3,2,4,8,2
3,3,0,3,2
3,3,0,5,2
3,3,0,8,2
3,3,1,3,4
3,3,1,5,4
3,3,1,8,4
3,3,2,3,4
3,3,2,5,4
3,3,2,8,4
3,3,3,3,4
3,3,3,5,4
3,3,3,8,4
3,3,4,3,2
3,3,4,5,2
3,3,4,8,2

4,1,0,3,2
4,1,0,5,2
4,1,0,8,2
4,1,1,3,4
4,1,1,5,4
4,1,1,8,4
4,1,2,3,4
4,1,2,5,4
4,1,2,8,4
4,1,3,3,4
4,1,3,5,4
4,1,3,8,4
4,1,4,3,2
4,1,4,5,2
4,1,4,8,2
4,2,0,3,2
4,2,0,5,2
4,2,0,8,2
4,2,1,3,4
4,2,1,5,4
4,2,1,8,4
4,2,2,3,4
4,2,2,5,4
4,2,2,8,4
4,2,3,3,4
4,2,3,5,4
4,2,3,8,4
4,2,4,3,2
4,2,4,5,2
4,2,4,8,2
4,3,0,3,2
4,3,0,5,2
4,3,0,8,2
4,3,1,3,4
4,3,1,5,4
4,3,1,8,4
4,3,2,3,4
4,3,2,5,4
4,3,2,8,4
4,3,3,3,4
4,3,3,5,4
4,3,3,8,4
4,3,4,3,2
4,3,4,5,2
4,3,4,8,2

5,1,0,3,2
5,1,0,5,2
5,1,0,8,2
5,1,1,3,4
5,1,1,5,4
5,1,1,8,4
5,1,2,3,4
5,1,2,5,4
5,1,2,8,4
5,1,3,3,4
5,1,3,5,4
5,1,3,8,4
5,1,4,3,2
5,1,4,5,2
5,1,4,8,2
5,2,0,3,2
5,2,0,5,2
5,2,0,8,2
5,2,1,3,4
5,2,1,5,4
5,2,1,8,4
5,2,2,3,4
5,2,2,5,4
5,2,2,8,4
5,2,3,3,4
5,2,3,5,4
5,2,3,8,4
5,2,4,3,2
5,2,4,5,2
5,2,4,8,2
5,3,0,3,2
5,3,0,5,2
5,3,0,8,2
5,3,1,3,4
5,3,1,5,4
5,3,1,8,4
5,3,2,3,4
5,3,2,5,4
5,3,2,8,4
5,3,3,3,4
5,3,3,5,4
5,3,3,8,4
5,3,4,3,2
5,3,4,5,2
5,3,4,8,2

Review of Industrial Organization **11:** 533–550, 1996.

Using Relative Profit Incentives to Prevent Collusion

CARL LUNDGREN*
1212 W. Jefferson, Apt. A, Springfield, IL 62702, U.S.A.

Abstract. This paper describes a new economic method for preventing oligopoly collusion. The method eliminates incentives for collusion by making managerial compensation depend on relative profits rather than absolute profits. This alteration of managerial incentives sets up a zero-sum game among the firms in an industry, yielding the result that firms no longer have incentive to collude, either actually or tacitly, with regard to prices or outputs. The method also ameliorates the imperfectly competitive outcomes which can result from even noncooperative oligopoly interactions.

Key words: Oligopoly, collusion, relative profits, zero-sum game, managerial incentives.

Introduction

The purpose of this paper is to present an alternative method for preventing collusion.[1] The method eliminates incentives for both actual and tacit collusion, and ameliorates the imperfectly competitive outcomes which can result from even noncooperative oligopoly interactions. The method prevents exploitation of oligopoly power, but is not a general cure for the market power problems of either strict monopoly or monopolistic competition.

Section I introduces and verbally describes the basic method of providing relative profit maximizing incentives for owners and managers of business firms. Section II reviews some related literature. Section III illustrates the method using a particular mathematical example. Section IV discusses some practical concerns related to implementing the method. Section V focuses on how firm owners can be prevented from making management stress absolute profits over relative profits. Section VI concludes. Three mathematical appendices derive: (A) the optimal weighting of rival firms' profits under a relative profit incentive scheme; (B) short-run equilibrium; and (C) Bertrand equilibrium for differentiated products.

* The author would like to thank numerous individuals for their comments on previous versions of this paper.

[1] NOTICE OF PATENT PENDING: This paper describes a method of economic regulation for preventing collusion upon which the author and inventor has applied for a patent. A patent on this invention, if such should be granted, would only restrict actual use of the described invention; it would not restrict in any way the verbal or written discussion, description, or criticism of that invention.

I. Basic Method

In an industry structure with only a few firms, collusion is a serious possibility, even when it is illegal. Tacit collusion, which does not require illegal communication among conspirators, can also occur.

The basic concept which underlies this proposed method is the perception that causing managers of firms to participate in a zero-sum game, or its equivalent or near-equivalent, will hinder or prevent cooperation or collusion among the managers of different firms. In a zero-sum game it is possible for one firm's manager to gain only if another firm's manager loses, since there is only a fixed quantity of rewards to go around. In a nonzero-sum game it is frequently possible for everyone to gain through cooperation (collusion) as opposed to noncooperation, since cooperation may increase the total quantity of rewards available to go around.

A zero-sum game for industry may be instituted by forcing firms as a whole to participate in a zero-sum game and/or by arranging zero-sum compensation arrangements for the managers of different firms. When the goal of firms is maximizing profits, instituting a zero-sum game in profits means that firms are motivated to maximize relative profits rather than absolute profits. That is, firms attempt to maximize the difference of their own firm's absolute profits relative to an average of other firms' absolute profits. Alternatively and equivalently, firms attempt to maximize the difference of their own firm's absolute profits relative to the average absolute profits of a group of competing firms, of which group the firm is a member.

A good way to institute a zero-sum game among firms in an industry is by motivating managers to seek relative profits rather than absolute profits. The usual way to motivate managers to pursue a particular goal is to pay managers in accordance with success in achieving that goal. If the goal is to maximize absolute profits, managers should expect to receive more compensation, the higher profits turn out to be. By altering the rules for managerial compensation in the appropriate way, we can make sure that managers are motivated to maximize relative profits rather than absolute profits.

The key to understanding this method rests upon the seemingly trivial observation that successful collusion increases the absolute profits of firms, but does not increase the relative profits of firms. When firms collusively raise prices, the relative profits of each firm cannot increase on average. Only one of two things can happen: Either (1) absolute profits of each firm rise equally and relative profits of each firm stay the same, or (2) the absolute profits of each firm do not rise equally, in which case some firms gain relative profit and some firms lose relative profit. If the second case holds true, any firm which loses relative profit from the collusive agreement will want to cheat (assuming it seeks relative profit), since it gains relative profit in the short run by cheating and it gains relative profit in the long run by breaking up the collusive agreement. If the first case holds true, no firm gains relative profit by maintaining the collusive agreement in the long run, and every firm gains relative profit by cheating in the short run. In a relative profit maximizing industry there is no incentive for all the firms to enter into or maintain any collusive agreement. Competitive behavior must result.

Setting up a zero-sum game in profits does not in any way require placing any cap or limitation on the amount of absolute or relative profit which any individual firm may earn. Rather, there is simply a definitional change in the type of profit which a firm or firm manager is expected to maximize. The main difference between absolute profit maximizing (APM) firms and relative profit maximizing (RPM) firms is that RPM firms are not motivated to collude. In the absence of collusion, absolute profit and relative profit are very similar. RPM firms are just as strongly motivated as APM firms to seek other sources of profit, such as reducing costs of production or improving product quality. RPM firms are not deliberately inefficient nor do they try to slow down technical progress. They merely refuse to collude, even tacitly.

Government is assumed able to observe costs and revenues *ex post*, but is not assumed able to observe either demand curves or cost functions. The proposed regulation is not heavy-handed. Price controls, profit controls, central commands, and the like are no part of the proposal. Under the relative-profit scheme of regulation, firms are perfectly free to try to make as much profit as they can, set whatever prices they wish, sell whatever products they wish, and to enter or exit industries and product lines as they please. Application of the RPM regulatory scheme need not extend beyond those firms which are most likely to collude (i.e., the largest firms within an oligopoly industry). Competitive industries, of course, do not need to be included (though no harm would come if they were).

II. Review of Some Related Literature

Only in Donaldson and Neary (1984) does there first appear a suggestion that the principles of relative profit maximizing might be put to practical use by altering the incentives of firms or managers. Donaldson and Neary suggest that relative profit maximizing managers in a "socialist industry" composed wholly of government-owned firms can achieve efficient outcomes with a minimum of administrative supervision by a central planner. They also prove numerous game theory propositions in this connection. Although they indirectly allude to the anti-collusive features of the incentive scheme, they never directly state this property outright. Consequently, they appear to have overlooked the possibility of extending the scheme to prevent collusion in privately-owned or "capitalist" industries. Also, they appear to impose the unwarranted restriction that each manager must be paid dollar-for-dollar for each dollar of relative profit which a firm earns (p. 102).

Two basic propositions in the Donaldson and Neary (1984) paper are worth special mention. The first is that RPM firms producing multiple or joint products will tend to produce at minimum cost and price efficiently (pp. 104–5, 109–10). This means that the incentive scheme is capable of being applied, not only to single-product firms and industries, but also to multi-product firms and industries.

Secondly, RPM firms, unlike their APM counterparts, have little or no strategic incentive to increase their market shares in a cost-inefficient manner by installing excess capital (pp. 105, 107–9).

The theoretical suggestion that firms with absolute profit incentives might under certain (presumably rare) circumstances try to behave as if they desired to maximize relative profits appears to have been made as early as 1960. Bishop (1960), describing the alleged "warfare" of oligopolists in the absence of collusion, Shubik and Levitan (1980), describing "beat-the-average" games, and Jones (1980), describing the outcome of a classroom game, each derive first-order conditions for a constant-sum game in relative profits. Jones also derives second-order conditions. Two reasons are suggested for such behavior: (1) Businessmen might be naturally rivalrous, caring more about relative position than absolute position, or (2) businessmen may be carrying out threats in order to elicit more favorable collusive agreements from their rivals in the future. These three works do not suggest any practical application for the mathematical principles of relative profit maximizing.

Although they do not anticipate the present subject matter, several other papers are worth mentioning. Holmstrom (1982) and Aron (1988) explore the use of relative performance evaluations for the quite distinct purpose of attempting more accurate evaluations of managerial performance. Gibbons and Murphy (1990) ask whether, in fact, managers tend to be paid according to relative performance. Fouraker and Siegel (1963) and Vickers (1985) consider relative profit goals and incentives of a different type, namely maximization of the difference between absolute profits of own firm and *total* absolute profits of rival firms, rather than *average* absolute profits of rival firms. Fershtman and Judd (1987) and Sklivas (1987) also consider alternative managerial incentives, but not relative profit incentives. Shleifer (1985) and Tam (1988) describe what may be the best currently known alternatives for regulating oligopoly markets, aside from antitrust enforcement or structural reform. Both of these alternatives require the regulation of prices, whereas the present method does not.

In summary, none of the previous literature suggests that relative performance incentives can be used as a general method for preventing collusion.

III. An Illustrative Example

Let G be some statistic which describes something about a firm. If the firm's managers are rewarded for achieving higher levels of G, then maximizing G will be the firm's goal or objective.

For purposes of this example, assume that there are N ($N \geq 2$) identical firms. Each firm produces a single, homogenous product at a constant marginal cost of C. The market demand is linear, with $P = A - bQ$, where $Q = \Sigma Q_i$ and Q_i is firm output. Assume that each firm pursues an identical goal function, which has a coefficient of unity in own firm profits and a coefficient of W in rival firm profits. The goal function for firm i and rival firm(s) j looks as follows:

$$\begin{aligned} G_i &= \pi_i + W \sum_{j \neq 1} \pi_j \\ &= \pi_i + W(N-1)\pi_j \\ &= (PQ_i - CQ_i) + W(N-1)(PQ_j - CQ_j) \end{aligned} \tag{1}$$

If $W = 0$, then the firm's goal is simply to maximize its own economic profit. This is the absolute profit maximizing (APM) goal. If $W = 1$, then the firm has a joint profit maximizing (JPM) goal. If all N firms have JPM goals, the industry will surely collude. On the other hand, if $W = -1/(N-1)$, then the firm has a relative profit maximizing (RPM) goal. The RPM goal is calculated by starting with own firm profits and subtracting off a weighted average of the $N - 1$ rival firm profits.

If we assume noncooperative behavior and Cournot conjectures, firm i maximizes its goal function by choosing Q_i such that:

$$\begin{aligned} \partial G_i/\partial Q_i &= (P - C) + (\partial P/\partial Q_i)[Q_i + W(N-1)Q_j] = 0 \\ &= (A - bQ_i - b(N-1)Q_j - C) \\ &\quad -b(Q_i + W(N-1)Q_j) = 0 \end{aligned} \tag{2}$$

For a symmetric, noncooperative equilibrium, assume that $Q_n = Q_i = Q_j$. Define $W_M = 1 + W(N-1)$. We can calculate the following quantities, price-cost margins, absolute profits, and goal fulfillments for each firm:

$$\begin{aligned} Q_n &= (A - C)/[b(N + W_M)] \\ P_n - C &= (A - C)W_M/(N + W_M) \\ \pi_n &= (A - C)^2 W_M/[b(N + W_M)^2] \\ G_n &= (A - C)^2 W_M^2/[b(N + W_M)^2] \end{aligned} \tag{3}$$

Now, assume instead that each firm pursues a collusive ("monopoly") equilibrium, in which each firm attempts to maximize its goal function under the assumption that all firms cooperate by setting the same level of output (Q_M) and receiving the same level of profit (π_M). The goal function takes the form:

$$\begin{aligned} G_M &= [1 + W(N-1)]\pi_M \\ &= W_M \pi_M \\ &= W_M(PQ_M - CQ_M) \end{aligned} \tag{4}$$

When $W > -1/(N-1)$, $W_M > 0$, so that joint goal fulfillment is equivalent to maximizing joint absolute profits. When $W < -1/(N-1)$, $W_M < 0$, joint goal fulfillment requires the minimization of joint absolute profits, or the maximization of joint losses. When $W = -1/(N-1)$, $W_M = 0$, we have a zero-sum game in relative profits. When $W_M = 0$, collusion can in no way improve the sum of relative profits for all N firms, since these must always add to zero. When $W_M = 0$, there is no incentive for all N firms to collude either to raise prices or to lower prices from the prices that would exist in a noncooperative equilibrium.

In what follows, assume that $W \geq -1/(N-1)$, so that $W_M \geq 0$ and joint maximization of absolute profits is a (weakly) plausible goal of collusion. (When $W_M = 0$, firms are collectively no better off, but neither are they collectively worse off, from collusion.) Then the collusive equilibrium has the following solution:

$$\begin{aligned}
\partial G_M/\partial Q_M &= W_M(P-C) + (\partial P/\partial Q_M)W_M Q_M = 0 \\
&= W_M(A - bNQ_M - C) - bNW_M Q_M = 0 \\
Q_M &= (A-C)/[2bN] \\
P_M - C &= (A-C)/2 \\
\pi_M &= (A-C)^2/[4bN] \\
G_M &= (A-C)^2 W_M/[4bN]
\end{aligned} \tag{5}$$

We now consider the one-period incentive for a firm to cheat on a collusive agreement. This can be calculated under the assumption that a single firm chooses its output to maximize its own goal function, taking as given that rival firms choose the agreed-upon collusive output level:

$$\begin{aligned}
G_C &= \pi_i + W(N-1)\pi_j(Q_j = Q_M) \\
&= (PQ_i - CQ_i) + W(N-1)(PQ_M - CQ_M)
\end{aligned} \tag{6}$$

This has solution:

$$\begin{aligned}
\partial G_C/\partial Q_i &= (P-C) + (\partial P/\partial Q_i)(Q_i + W(N-1)Q_M) = 0 \\
&= (A - bQ_i - b(N-1)Q_M - C) - b(Q_i + W(N-1)Q_M) = 0 \\
Q_c &= (A-C)(N+2-W_M)/[4bN] \\
P_C - C &= (A-C)(N+W_M)/[4N] \\
\pi_C &= (A-C)^2(N+2-W_M)(N+W_M)/[16bN^2] \\
\pi_j &= (A-C)^2(N+W_M)/[8bN^2] \\
G_C &= (A-C)^2(N+W_M)^2/[16bN^2]
\end{aligned} \tag{7}$$

The reward to a firm which colludes in a repeated game with its rivals is:

$$R_M = G_M + \sum_{t=1}^{\infty} G_M/(1+r)^t = G_M + G_M/r \quad (8)$$

The value of r depends not simply on the cost of capital and risk premia, but also includes the probability that collusion may break down, perhaps because of industry changes or government intervention. The length of the time period, t, depends on the time it takes for rivals to discover that cheating has occured, after which collusion breaks down. The shorter the time period needed to detect cheating, the lower the value of r. The reward to a firm which cheats in period $t = 0$ and sees the noncooperative equilibrium in subsequent periods is:

$$R_C = G_C + \sum_{t=1}^{\infty} G_n/(1+r)^t = G_C + G_n/r \quad (9)$$

$R_M > R_C$ (so that collusion is sustainable) whenever the collusion/cheating ratio shown below exceeds r:

$$(G_M - G_n)/(G_C - G_M) = 4NW_M/(N + W_M)^2 > r \quad (10)$$

For a given N within the relevant range, this ratio reaches its maximum value of 1 when $W_M = N$ (JPM) and reaches its minimum value of 0 when $W_M = 0$ (RPM). The ratio rises monotonically when W_M increases from 0 to N. As might be expected, when $W_M = 1$ (APM), this ratio falls (i.e., collusion is harder to sustain) when the number of firms (N) increases.

To summarize, when firms are given RPM incentives and placed in a zero-sum game, the incentive to collude is eliminated, but the incentive to cheat on collusion is maintained. No collusive agreement can benefit all firms in a zero-sum game, and any such agreement would in any case be subject to overwhelming incentives for most or all firms to cheat. This was shown verbally in Section I and is illustrated in this section using a particular mathematical model. The details of a mathematical model can be varied endlessly, but the qualitative conclusion will always be the same, given the verbal proof in Section I.

IV. Practical Implementation

Economists traditionally present theory and presume (sometimes unrealistically) that the manner of its practice will be immediately apparent. With respect to many practical concerns which some economists and laymen have raised, some brief answers are indicated below.

1. *Would government regulators need extensive and expensive data to enforce the proposed scheme?* No. The only data needed are data on costs, revenues, profits, and managerial compensation. Since this data must be collected in any case, either

by government for tax purposes, or by accountants as a prudent way for managers and stockholders to keep tabs on a firm's activities and cash flows, it follows that the method can be implemented at little or no extra cost.

This data is readily observable, so governmental omniscience is not required to implement the method. In particular, it is not assumed that government can observe either cost functions or demand curves, nor is it assumed that government can calculate optimal prices, profits, or output levels. Hence, the RPM method can be practiced, even if government is unable (because of information limitations) to set optimal prices or quantities directly.

2. *Would the use of accounting data to measure costs, revenues, and profits cause economic distortions?* Perhaps, but a more relevant question might be, would such distortions be any greater under RPM than under APM? The purpose of the method is to prevent collusion, not to calculate true economic costs or profits. Even under current arrangements, inability to measure true economic cost prevents stockholders from motivating managers with proper incentives to maximize absolute profits. Whatever may be (for motivational purposes) the most accurate way to measure absolute profits can also be used as a good way to measure relative profits. Regardless of whether profits are calculated using accounting data or other imperfect data, collusion will be prevented, and there is unlikely to be any significant incremental effect in causing additional misallocation of resources.

3. *How does one measure "relative profit"?* Aside from accounting measures, one way to estimate absolute profit is to look at changes in the value of a firm's total outstanding stock over a period of time and make adjustments at an appropriate interest rate for dividends paid or new stock shares issued over the same period of time. Since changes in both short-term and long-term profit potential affect the firm's value, this method of ascertaining profit gives managers the least incentive to manipulate accounting procedures, or to manipulate events in response to mistaken accounting rules. To calculate relative profits by this method, one simply looks at the change in value for one firm and subtracts off a weighted average of the change in value for rival firm(s).

An alternative method for measuring relative profit makes use of a new forecasting method, described by Lundgren (1995). This method provides efficient incentives for unbiased human forecasts of any variable value, including the future absolute profits or relative profits of any firm or any subcomponent of a firm, and such forecasts can be made as free of accounting biases as stock values. An advantage of the forecasting method is that it can be used to separate out the industry-specific profits of a conglomerate operating in several industries.

4. *How does one apply the relative profit concept to industries which contain multi-industry conglomerates?* Most multi-industry conglomerates adopt the multi-division form of organization, in which each industry division is operated essentially as a separate profit center, with separate accounting for each industry of operation. If the conglomerate operates in unrelated industries, there is unlikely to be any economy of scale or scope that would be wasted if the conglomerate

were required to break itself up into single-industry parts. If a break-up is deemed undesirable because of economies of scale or scope, and if it is infeasible to issue separate securities for each industry subsidiary of the firm, then one can either use accounting techniques or use the forecasting technique described in Lindgren (1995). If a firm simply produces multiple (but closely related) products, the firm is best understood as producing in a single industry – a circumstance which requires no special treatment, as shown in Donaldson and Neary (1984, pp. 104–5, 109–110).

5. *How does one define the "market" or "industry" for purposes of imposing the zero-sum game?* Since it is not the purpose of the scheme to determine legal culpability for monopolization, but simply to eliminate incentives for collusion, it is not necessary to answer the tricky question of how broadly or narrowly the market should be defined. It is generally preferable to define the industry/market rather narrowly, so that only a very few, very similar firms are placed in each zero-sum game. That is, if there is a broadly defined industry with several firms, it is generally preferable to impose more than one zero-sum game on the several firms, by grouping the firms into more narrowly defined sub-industries, and imposing a zero-sum game on each of the smaller groups. Unlike under current antitrust law, it is not necessary to inquire whether more distantly related firms are actually part of the same "market".

6. *How does one sustain incentives for innovation and technological progress?* Innovations may be either costless or costly, and may be either patentable or unpatentable. If innovations are costless, we may presume that relative profit maximizers will adopt them, since profit maximization implies cost minimization. If innovations are costly, but patentable, the patent law provides incentive for innovation. Since RPM incentives are designed to induce competition, they should not be applied to situations where monopoly, and hence absolute profit maximizing, is the preferred public policy. Fortunately, both absolute profit and relative profit are measured in compatible money units, so there is nothing to prevent the institution of APM incentives for patented activities and RPM incentives for unpatented activities, even with respect to the same manager in the same firm.[2]

If innovations are costly, but unpatentable, RPM firms still have an incentive to reduce costs, if gains from innovation can be captured for a period of time until competitors follow suit. This incentive is proportional to firm output. The conventional Schumpeterian "wisdom" that a competitive industry is less innovative than an oligopolistic industry confounds the influence of firm size with the competitive/noncompetitive nature of firm interaction. It is mainly the size of firm output, not the size of a collusive price-cost margin, which determines the size of the incentive to reduce unit costs.

[2] There are various ways this can be done. For example, if firm A has a patent and firm B is a rival, any royalty payment from firm B to firm A would not be counted against either firm A or firm B in the calculation of relative profits. A complete exposition would require a separate paper.

7. *How does one prevent RPM industries from sustaining chronic losses?* Chronic losses would occur only if marginal cost lies consistently below average cost for a particular industry. In such case, the industry can be made viable by offering an industry lump-sum subsidy in the exact amount of the industry's economic losses. Lump-sum subsidies may be distributed equally to all firms in a zero-sum group without affecting relative profits, and hence without inducing behavior to manipulate the size of the subsidy. Financing the subsidy through general revenues yields marginal cost pricing. Financing through a special industry tax yields average cost pricing.

8. *How does one ensure that RPM firms do not sabotage rival firms' operations?* Since relative profits rise when rival firm profits fall, there is arguably an increased incentive to sabotage rival firm operations. An increased incentive to cause sabotage need not imply a significant increase in actual sabotage. A situation of mutual sabotage can only arise if legal penalties are very weak, since rival firms have incentive to investigate, report, and prosecute sabotage activities which reduce their levels of profit.

Nevertheless, even if we were to suppose that serious sabotage problems would arise from an unmodified RPM incentive scheme, it is possible to modify the incentive scheme slightly so as to eliminate the sabotaging incentives. This modification would require a deduction in managerial compensation which offsets (or further penalizes) any gain in managerial compensation resulting from any gain in relative profits due to sabotage occurring in rival firms, even if legal culpability for the sabotage cannot be established. In other words, one may convert the zero-sum game into a negative-sum game, if sabotage is observed. (One can apply the same reasoning to lawsuits.)

9. *How does one ensure that corporate managers will not evade the regulation of salary policies?* The regulation of managerial compensation has nothing to do with the total amount of the salary and bonuses, but only the methods of their calculation. Even if we suppose that the value of relative profits is lower, on average, than the value of absolute profits, the noncontingent salary component of a manager's compensation can always be raised to compensate. No reduction in the average levels of managerial compensation is required. For the same level of risk and expected compensation, managers do not care whether bonuses are contingent on relative profit or absolute profit.

10. *Does the scheme represent an unwarranted intrusion into managerial compensation policies which have heretofore been unregulated?* The proposal does not actually require government to *determine* the managerial incentive schemes. It simply requires that the contingent part of any managerial incentive scheme must be based on relative firm performance, rather than absolute firm performance. In order to prevent the incentive for managerial collusion, it is not necessary that government determine the overall level of managerial compensation, nor is it necessary that government determine and implement any particular method for measuring

relative firm performance. The minor intrusion, if it be such, is justified by the important public purpose at stake: Preventing collusion.

11. *How does one prevent collusion among managers to reduce managerial effort levels?* Instituting a zero-sum game in managerial income does not mean instituting a zero-sum game in managerial effort levels, so collusion to reduce managerial efforts is at least conceivable. However, collusion to reduce effort levels is not a serious threat, since a) managers of firms typically work in separate locations, and b) the work of managers consists mainly of mental efforts. Therefore, since managerial effort is essentially unobservable, any agreement to reduce effort levels cannot be easily monitored or enforced by colluding managers.

However, simply for argument's sake, suppose that managerial effort is actually (at least partly) observable. For example, suppose effort can be measured based on hours spent "on the job". In that case, one can pay managers based both on absolute effort and on relative performance. If the compensation rate for effort is made high enough, managers will no longer have incentive to collude to reduce effort levels, even if such collusion could be made perfectly enforceable.

12. *How does one ensure that firm owners will not find ways of making management stress absolute profits over relative profits?* This is the subject of the next section.

V. Owners, Managers, and Incentives

There are at least two ways of instituting relative profit incentives for firms. First, top management (including the board of directors) can be given long-term compensation contracts based on relative performance. Secondly, one can impose the zero-sum game in profits on whole firms (owners), and not just managers. Government may adopt only the first set of measures, only the second set, or both sets simultaneously.

The second method can be implemented by 100% taxing (subsidizing) the combined economic profits (losses) of an industry and allocating the tax (subsidy) equally to each firm. The tax (subsidy) would be on *industry* profits, *not* individual *firm* profits. As a result of the industry tax/subsidy scheme, after-tax profits to owner-shareholders are equivalent to pre-tax relative profits, which means owners will try to maximize relative profits rather than absolute profits. Although this method appears economically viable, it may not be politically palatable, given the potential for redistributions of income between stockholders and the government.

In firms or industries where owners and managers are one and the same, these two methods are essentially equivalent. No choice is possible. However, most important oligopoly industries are probably composed of large corporations which maintain a separation between ownership and direct managerial control. This well-known aspect of the internal structure of the modern corporation presents an interesting avenue by which government can enforce antitrust policy. Rather than impose relative profit incentives directly upon owners through taxes and subsidies, government can influence firm behavior simply by altering the incentives of management.

However, if owners are not made the direct subjects of taxes and subsidies which impose relative profit incentives, this raises the issue of whether absolute wealth maximizing stockholders can somehow reimpose APM incentives on RPM managers. The current state of corporate affairs is that managers, not stockholders, basically control the large corporation. Managers effectively appoint the boards of directors, to whom they are ostensibly responsible. Indirectly, through their choice of board members, managers set their own salaries. Managers have no incentive to change this state of affairs.

Stockholders are numerous and dispersed. Individually, most stockholders do not have enough votes to unseat management. Obtaining collective action to unseat management requires significant expense, which most stockholders find too costly to undertake. Controlling management is a "public good" for all stockholders, which most stockholders find rational to "free ride" upon by not attempting to provide it. The only stockholders who might have an incentive to undo management policies are the largest or principal stockholders. The remaining stockholders are of no consequence, except as voters who might side with the principal stockholder in any fight against management.

Consider, therefore, an industry in which the top two, three, or four firms have been placed into a zero-sum game in terms of managerial incentives. Each firm has a different principal owner. If the same person or entity is a principal owner in two or more of the top firms in an oligopoly industry, this should be regarded as an antitrust violation, just the same as interlocking directorates are so regarded. Hence, we assume different principal owners. Any conspiracy to undo the relative incentive scheme must involve the principal owners, since the managers themselves have no such incentive.

To be effective, the conspiracy must convert all or most managers from relative profit goals to absolute goals. To convert only one manager to absolute profit goals would not generate the kind of collusion among business firms which could substantially raise prices and profits, and thereby make the conspiracy (with its attendant risks) worthwhile from a private perspective. The conspiracy must therefore involve the principal owners of different firms acting in combination. The principal owners, being already wealthy, will not rationally risk jail time simply to increase their wealth still further. Suppose, nevertheless, that the principal owners attempt a conspiracy. What means would they use to influence management?

There are basically only two avenues by which the principal owners might try to influence management: compensation and employment. Either avenue may be pursued overtly or covertly.

First consider overt operations. The principal owner persuades stockholder voters to alter the conditions of employment or compensation. For example, the principal owner might use the annual stockholder meetings to directly hire or fire the manager, according to whether the manager pursued or failed to pursue collusive policies alongside other firms. Alternatively, the annual meetings might be used to raise or lower the base salary for future employment in a manner designed to undo the relative profit incentives paid in previous years.

Use of the annual stockholder meetings for either purpose would be an unusual or abnormal business practice. Use of the annual meetings for these purposes by an RPM firm would clearly be a suspect practice, prompting an antitrust investigation. More simply, use of a stockholder meeting to directly determine managerial employment or compensation might be made a *per se* antitrust violation, when performed by an RPM firm.

Thus, the following is recommended for instituting RPM incentives on management: Both the managers and the boards of directors are given long-term contracts containing relative performance incentives, which are not altered from year to year in a manner that might allow an owner to undo the RPM incentives. Any part of the compensation (including stock holdings or stock options) which is contingent on the firm's performance must be based on relative performance, not absolute performance. All compensation and compensation arrangements of managers and directors of RPM firms are disclosed to the antitrust authorities. Managerial employment is determined by the board of directors (all of whom are paid according to RPM incentives), not by either stockholders or principal owners, unless the principal owners have been converted to RPM incentives. The directors have overlapping terms, and are not all elected at once.

In situations where a principal owner (or other stockholder) wishes to have an active role in management or on the board of directors of an RPM firm, such owner or stockholder must have his stockholdings converted into assets which provide RPM incentives. This can be accomplished either by shorting the stock of rival RPM firms, and/or by imposing a tax/subsidy on the stockholder which mimicks the change in value of the stock in rival RPM firms. If the principal stockholder desires to be a passive investor, this change in incentive is not required.

Consider now possible covert operations. Assume that the principal owner has not acknowledged any active interest in the corporation, and has not been converted to RPM incentives. How can a principal owner with APM incentives undo the RPM incentives of firm managers? There are only two possibilities: threats and bribes. Threats are particularly likely to be reported to the antitrust authorities, could result in extra jail time, and will presumably not be resorted to. This leaves bribery.

The rich stockholder may choose to bribe either the manager or the directors. The manager might be bribed to behave collusively. The directors might be bribed to hire and fire managers based on willingness to collude. Bribing the directors is likely to be cheaper, but also less effective and more likely to be reported. Even if successful, bribing the directors to fire a manager is particularly likely to be reported by the manager. This leaves only bribing the manager directly.

A conspiracy by principal owner(s) will not stop at trying to institute APM incentives on RPM managers. Rather, the rational goal would be to attempt to institute joint-profit maximizing (JPM) incentives on the managers. Bribing the managers of all the important firms in the industry to institute JPM incentives is the only procedure that would guarantee collusion. Re-instituting APM incentives merely provides the opportunity for collusion, but does not guarantee its occurrence.

The possibility that principal owner(s) might bribe the manager(s) of APM firms to collude, or provide them with JPM incentives exists even today. Yet one rarely (or never?) hears of principal owner(s) attempting to bribe or covertly pay firm managers in this manner. If such behavior does not happen when firm managers are paid according to APM incentives, why should it happen if managers are paid according to RPM incentives?

In short, an illegal conspiracy of stockholders to re-impose absolute profit incentives onto firm managers is unlikely. The small stockholder has little influence and insufficient incentive to launch such a conspiracy. The large stockholder is too wealthy to want to risk jail time. Any such conspiracy would have to be explicit (and therefore detectable), not merely tacit.

VI. Conclusion

It has been shown that institution of a zero-sum game among a group of firms by means of relative profit maximizing incentives is capable of reducing or eliminating incentives for firms to collude, either actually or tacitly. This mild change in managerial incentives can be imposed at essentially zero public or private cost, yet it reaps potentially huge benefits.[3]

Appendix A. Optimal Weighting of Goal Functions

This appendix derives the conditions for goal functions needed to achieve a zero-sum game with desirable long-run properties. Suppose there are N ($N \geq 2$) firms in an industry. Let $\pi_1, \pi_2, \ldots, \pi_N$ be the profits earned by these firms, and let G_1, $G_2, \ldots, G_N$ be the goal functions for these firms. Let w_{ij} be the weight placed on firm j's profits in firm i's goal function, and let K_i be an arbitrary constant which adjusts firm i's goal satsifaction upwards or downwards (e.g., a fixed salary component in managerial pay). Goal functions which are linear in profits have the form:

$$G_i = \sum_{j=1}^{N} w_{ij}\pi_j + K_i \qquad \text{(A.1)}$$

for all $i \in [1, N]$, where all w's and K's are fixed constants.

The zero-sum conditions require:

$$\sum_{i=1}^{N} w_{ij} = 0, \qquad \text{(A.2)}$$

[3] One famous estimate of the deadweight cost of monopoly power (including oligopoly) is between 1/2% and 2% of G.N.P. (Scherer and Ross, 1990, p. 667)

Suppose each firm sells a standardized product and that price depends solely on industry output: $P = p(Q)$, where $Q = \Sigma Q_i$. Assume further (which is likely in the long run) that cost functions are identical for each firm: $TC_i = C(Q_i)$. Hence, $\pi_i = PQ_i - C(Q_i)$, so that we obtain:

$$G_i = \sum_{j=1}^{N} w_{ij}(PQ_j - C(Q_j)) + K_i \tag{A.3}$$

Define $a_{ji} = \partial Q_j/\partial Q_i$ as any arbitrary conjecture which firm i entertains about the reaction function of firm j ($a_{ii} = 1$). Thus:

$$(\partial G_i/\partial Q_i) = \sum_{j=1}^{N} a_{ji}w_{ij}[P - C'(Q_j)] + p'(Q)\sum_{j=1}^{N} w_{ij}Q_j = 0 \tag{A.4}$$

Assuming that N firms in the industry is a given and that marginal cost is increasing, then the optimal industry outcome occurs only when firm outputs are identical ($Q_j = Q/N$) and price equals marginal cost $[P - C'(Q_j) = 0]$ for each firm. These conditions are met only when the weights on profits meet the following conditions:

$$\sum_{j=1}^{N} w_{ij} = 0 \tag{A.5}$$

These N conditions for long-run industry optimality are in addition to the N conditions in (A.2) on weights needed to insure the zero-sum nature of the game. Short-run effects[4] not analyzed here may perhaps place additional restrictions on the optimal values for weights in the goal functions of firms.

Appendix B. Short-run Cost Differences

This appendix employs a 2-firm game to model the consequences of short-run cost differences between firms. Let $C_1(Q_1)$ and $C_2(Q_2)$ be the cost functions of firms 1 and 2, and let $a_{12} = \partial Q_1/\partial Q_2$ and $a_{21} = \partial Q_2/\partial Q_1$ be any arbitrary conjectures which each firm entertains about the reaction functions of the other firm. The problem itself may be stated thus:

$$G_1 = \pi_1 - \pi_2 = p(Q)Q_1 - C_1(Q_1) - p(Q)Q_2 + C_2(Q_2), \tag{B.1}$$

$$G_2 = \pi_2 - \pi_1 = p(Q)Q_2 - C_2(Q_2) - p(Q)Q_1 + C_1(Q_1). \tag{B.2}$$

[4] These short-run effects include possible incentives for pair-wise or subset collusion, or possible concerns about distribution of output among firms, if short-run costs differ significantly among firms and there are three or more firms in the same zero-sum game.

The first order conditions are:

$$\begin{aligned} \partial G_1/\partial Q_1 &= p'(Q)(1+a_{21})(Q_1-Q_2)+p(Q)(1-a_{21}) \\ &\quad -C_1'(Q_1)+a_{21}C_2'(Q_2)=0 \end{aligned} \tag{B.3}$$

$$\begin{aligned} \partial G_2/\partial Q_2 &= p'(Q)(1+a_{12})(Q_2-Q_1)+p(Q)(1-a_{12}) \\ &\quad -C_2'(Q_2)+a_{12}C_1'(Q_1)=0 \end{aligned} \tag{B.4}$$

It can be shown that the only solution which satisfies Equations (B.3) and (B.4) has the form:

$$p(Q)=\frac{C_1'(Q_1)+C_2'(Q_2)}{2} \tag{B.5}$$

$$(Q_1-Q_2)=\frac{C_1'(Q_1)-C_2'(Q_2)}{2p'(Q)} \tag{B.6}$$

Equation (B.5) tells us that price will be set equal to the average of the marginal costs of the two firms. Equation (B.6) tells us that the more efficient firm will produce more output than the less efficient firm, since $p'(Q)<0$.

Appendix C. Differentiated Products

This appendix models what happens when products are differentiated and firms compete in prices (Bertrand conjectures). Let the demand structure for the two firms be defined in terms of quantities demanded as a function of two prices: $Q_1 = Q_1(P_1,P_2)$ and $Q_2=Q_2(P_1,P_2)$. It is reasonable to assume $(\partial Q_j/\partial P_j)<0$ and $(\partial Q_i/\partial P_j)>0$ for all i and $j, j\neq i$. The problem is stated thus:

$$\begin{aligned} G_1=\pi_1-\pi_2 &= Q_1(P_1,P_2)P_1-C_1(Q_1(P_1,P_2)) \\ &\quad -Q_2(P_1,P_2)P_2+C_2(Q_2(P_1,P_2)) \end{aligned} \tag{C.1}$$

$$\begin{aligned} G_2=\pi_2-\pi_1 &= Q_2(P_1,P_2)P_2-C_2(Q_2(P_1,P_2)) \\ &\quad -Q_1(P_1,P_2)P_1+C_1(Q_1(P_1,P_2)) \end{aligned} \tag{C.2}$$

The first-order conditions are:

$$\begin{aligned} (\partial G_1/\partial P_1) &= Q_1+P_1(\partial Q_1/\partial P_1)-C_1'(Q_1)(\partial Q_1/\partial P_1) \\ &\quad -P_2(\partial Q_2/\partial P_1)+C_2'(Q_2)(\partial Q_2/\partial P_1)=0 \end{aligned} \tag{C.3}$$

$$(\partial G_2/\partial P_2) = Q_2 + P_2(\partial Q_2/\partial P_2) - C_2'(Q_2)(\partial Q_2/\partial P_2) - P_1(\partial Q_1/\partial P_2) + C_1'(Q_1)(\partial Q_1/\partial P_2) = 0 \tag{C.4}$$

We may rearrange these conditions as expressing the determinants of price-cost margins:

$$P_1 - C_1'(Q_1) = \frac{-Q_1 + [P_2 - C_2'(Q_2)](\partial Q_2/\partial P_1)}{(\partial Q_1/\partial P_1)} \tag{C.5}$$

$$P_2 - C_2'(Q_2) = \frac{-Q_2 + [P_1 - C_1'(Q_1)](\partial Q_1/\partial P_2)}{(\partial Q_2/\partial P_2)} \tag{C.6}$$

The corresponding conditions for APM firms with Bertrand conjectures are:

$$P_1 - C_1'(Q_1) = \frac{-Q_1}{(\partial Q_1/\partial P_1)} \tag{C.7}$$

$$P_2 - C_2'(Q_2) = \frac{-Q_2}{(\partial Q_2/\partial P_2)} \tag{C.8}$$

Suppose we have a situation where two monopolistically competitive firms satisfy equations (C.7) and (C.8) because both firms are maximizing absolute profits. Assume that both firms have positive price-cost margins. Now suppose that one (or both) of the firms is converted into being a relative profit maximizer. If firm one is so converted, its price-cost margin is reduced, because the difference between the right-hand-side terms in equations (C.5) and (C.7) is negative:

$$\frac{[P_2 - C_2'(Q_2)](\partial Q_2/\partial P_1)}{(\partial Q_1/\partial P_1)} < 0 \tag{C.9}$$

Similarly, if firm 2 is converted to relative profit maximizing (assuming firm 1 maintains a positive price-cost margin), it too will wish to reduce its price-cost margin by expanding output and lowering price. Bertrand competition is absolutely the most competitive behavior which it is reasonable to postulate about APM firms, yet RPM firms compete even harder.

References

Aron, Debra J. (1988) 'Ability, Moral Hazard, Firm Size, and Diversification', *Rand Journal of Economics*, **19**(1), 72–87.

Bishop, Robert L. (1960) 'Duopoly: Collusion or Warfare?', *American Economic Review*, **50**, 933–961.

Donaldson, David and Hugh Neary (1984) 'Decentralized Control of a Socialist Industry', *Canadian Journal of Economics*, **17**(1), 99–110.

Fershtman, Chaim and Kenneth L. Judd (1987) 'Equilibrium Incentives in Oligopoly', *American Economic Review*, **77**(5), 927–940.

Fouraker, Lawrence E. and Sidney Siegel (1963) *Bargaining Behavior*. New York: McGraw-Hill.

Gibbons, Robert and Kevin J. Murphy (1990) 'Relative Performance Evaluation for Chief Executive Officers', *Industrial and Labor Relations Review*, **43**, Special Issue, 30S–51S.

Holmstrom, Bengt (1982) 'Moral Hazard in Teams', *Bell Journal of Economics*, **13**(2), 324–340.

Lundgren, Carl (1995) 'Forecasting Incentives Based on Value Marginal Product', manuscript, June.

Jones, Michael (1980) 'Note on oligopoly: Rival behavior and efficiency', *Bell Journal of Economics*, **11**(2), 709–714.

Scherer, F. M. and David Ross (1990) *Industrial Market Structure and Economic Performance*, Third Edition. Boston, MA: Houghton Mifflin Company.

Shleifer, Andrei (1985) 'A Theory of Yardstick Competition', *Rand Journal of Economics*, **16**(3), 319–327.

Shubik, M. and R. Levitan (1980) *Market Structure and Behavior*. Cambridge: Harvard University Press.

Sklivas, Steven D. (1987) 'The Strategic Choice of Managerial Incentives', *Rand Journal of Economics*, **18**(3), 452–458.

Tam, Mo-Yin S. (1988) 'A Mechanism to Induce Ramsey Pricing for Natural Monopoly Firms', *International Journal of Industrial Organization*, **6**(2), 247–262.

Vickers, John (1985) 'Delegation and the Theory of the Firm', *Economic Journal*, **95**, Conference Papers Supplement, 138–147.