A SIMULATION OF THE EFFECTS OF FOUR INSECT PESTS ON RED PINE GROWTH AND YIELD IN MICHIGAN

By

Stephen Carl Westin

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Forestry

1987

ABSTRACT

A SIMULATION OF THE EFFECTS OF
FOUR INSECT PESTS ON
RED PINE GROWTH AND YIELD
IN MICHIGAN

By

Stephen Carl Westin

A model was developed to simulate the effects of white grubs (Phyllophaga spp., Coleoptera, Scarabaeidae), Saratoga spittlebug (Aphrophora saratogensis [Fitch]), pine root collar weevil (Hylobius radicis Buch.), and red-headed pine sawfly (Neodiprion lecontei [Fitch]) on growth and yield of red pine (Pinus resinosa Ait.). The model allows the user to set initial stand conditions, and choose one of the insects to simulate. Insect levels are randomly set. Effects of insect damage may be offset by implementing control measures. The stand may be thinned to varying levels at any time. An economic analysis is performed to compare costs and returns with and without insect management.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Carl W. Ramm, graduate committee chairman, for his guidance, patience, and friendship during the course of my graduate program. Thanks are due also to Dr. Gary Simmons, Dr. Douglas Lantagne, and Dr. Donald Dickmann for their ideas, support, and timely action when called upon. The completion of this thesis would not have been possible without their help.

Also, I would like to thank my family for their support and patience throughout this rather interesting experience.

Finally, I would like to thank Karen Stange for being there when I needed her. Her love has made my life much more meaningful.

TABLE OF CONTENTS

List of Tables....................................vi

List of Figures...................................vii

Introduction......................................1

Background Information............................3

Overview of the Model.............................7

Plantation Establishment..........................10

Red Pine Growth and Yield.........................13

- Red Pine Growth..............................13
- Diameter Distribution........................18
- Estimating Red Pine Volume...................19
- Estimating Red Pine Mortality................21

Region of Applicability for Insect Models.........22

White Grubs.......................................23

- White Grub Caused Mortality..................23
- Growth Loss Due to White Grubs...............26
- Control Measures and Efficacy................26

Pine Root Collar Weevil...........................27

- Simulating Pine Root Collar Weevil Impact on Red Pine.....................................29

Saratoga Spittlebug...............................31

Simulating Saratoga Spittlebug Impact on Red Pine....33
Red-Headed Pine Sawfly....36
Simulating Red-Headed Pine Sawfly Impact on Red Pine Volume....38
Economic Evaluation....40
Summary....41
List of References....43
Appendix A: A User's Guide For RPPEST....48
Appendix B: A Program Listing of RPPEST....79

LIST OF TABLES

1. Lower and upper limits of red pine seedling mortality through year 2, for a given mean number of *Phyllophaga* per cubic foot of soil..24
2. Probability of heavy damage to red pine from root collar weevil, by hazard zone and distance to a weevil infestation.................28
3. Default costs for forest management activities and stumpage...................................54
4. Range of random values used in RPPEST simulations...................................56
5. Default costs for insect management activities in RPPEST.......................................57

LIST OF FIGURES

1. A Flowchart of RPPEST....8
2. Initial Stand Information....63
3. Initial Site Preparation and Economic Information....64
4. First Year Seedling Survival....65
5. Insect Caused Seedling Mortality....66
6. Insect Monitor and Control Costs....67
7. Stand Description Display At Age 20....68
8. Stand Description Display At Age 30....69
9. Stand Description after Thinning At Age 35....70
10. Thinning Parameter Change Menu....71
11. Stand Description Display At Age 40....72
12. Stand Description After Thinning At Age 45....73
13. Stand Description Display At Age 50....74
14. Stand 1 Volume Summary....75
15. Stand 2 Volume Summary....76
16. Final Economic Analysis....77

INTRODUCTION

Due to the length of most forest stand rotations, land managers spend a great deal of their careers gaining the experience and knowledge necessary to be successful in their management activities. For example, although heavy attacks by insects are often infrequent, the resultant damage may be serious. Because of lack of experience , a land manager may not know the most effective or economical way to deal with a serious insect outbreak when it occurs. Computerized growth and yield models which simulate insect impact allow managers to experiment with various management strategies for trees and insects, helping them gain necessary experience.

The purpose of the model described in this thesis and the associated computer program, RPPEST, is to serve as a training tool for land managers. The model simulates the impact of four insects on the growth and volume yield of red pine (*Pinus resinosa* Ait.). The insects are: white grubs (*Phyllophaga* spp., Coleoptera, Scarabaeidae), pine root collar weevil (*Hylobius radicis* Buch.), Saratoga spittlebug (*Aphrophora saratogensis* [Fitch]), and red-headed pine sawfly (*Neodiprion lecontei* [Fitch]). This model allows flexible stand definition as well as relative ease of use on a microcomputer. Economic analysis of various combinations of timber and insect management strategies should make the model a useful tool to help land managers gain experience within a shorter time frame and at a lower cost than that allowed by the real world.

Background Information

Red pine (*Pinus resinosa* Ait.) is a widely planted forest species in the Lake States region and many forest products are derived from it, including pulpwood, posts, utility poles, cabin logs, piling and sawtimber (Benzie, 1977). Because rotation lengths needed to produce these products often range from 60 to 120 years (Benzie and McCumber, 1983), land managers responsible for obtaining maximum benefit from red pine stands often do not know the results of their management prescriptions until late in their careers, if ever. As in most other professions, foresters gradually gain experience and knowledge as their careers progress. However, the long interval between the beginning and end of most forestry projects slows the accumulation of knowledge. Accelerating this learning process should favor more effective forest management.

Computer based growth and yield models allow forest managers to simulate various stand management scenarios and evaluate their results in a short of time, and with no risk. Prior to this time, at least four computerized growth and yield models were available for red pine in the Lake States: REDPINE (Lundgren, 1981), STEMS (Belcher et al., 1982), RPAL (Ramm and Miner, 1986), and TWIGS (Belcher, 1982).

REDPINE, briefly described by Lundgren (1981), is a stand level growth and yield model for red pine plantations or natural stands in the Lake States. The program forecasts growth beginning at any stand age for thinned or unthinned stands. If a stand is grown from the time of planting REDPINE makes no attempt to account for seedling mortality in the first years after planting. REDPINE allows wide latitude in the description of the stand in terms of number of trees per acre, initial basal area, site index, and thinning specifications. REDPINE was designed and written for mainframe computers, which somewhat limits access to it. RPAL (Ramm and Miner, 1986) is a BASIC language translation of REDPINE written for the IBM PC and compatible microcomputers.

STEMS is an individual tree growth and yield projection system for over thirty species in the Lake States (Belcher et al., 1982). As an individual tree model, STEMS requires that a list of trees and their characteristics be entered before projection begins, or the program will generate such a list if requested. STEMS allows implementation of various management systems, such as removal of cull trees and thinning from above or below (Brand, 1979). Simulation of mortality is accomplished using a mortality function based on individual tree diameter growth rate (Buchman, 1979). A microcomputer version of STEMS called TWIGS has been developed, which will increase public access to this growth projection system.

Attacks on forest stands by some insects may be infrequent but serious when they do occur. Other insects frequently cause low to moderate levels of tree damage or mortality and often go unnoticed. Forest managers need to understand the nature of these insect attacks, their impact on growth and yield, and the effect of possible control actions on stand growth and yield. A logical extension of computerized growth and yield models for forest stands would be the addition of simulated insect attack on the stands. Two computerized models are available for the Lake States region.

BANKSIANA is a computerized stand level growth and yield model for the Lake States which simulates the impact of white pine weevil and jack pine budworm on jack pine (Drapek, 1985). BANKSIANA allows several methods of jack pine reproduction with variable results. Insect attack is randomly determined and several control measures are available. Economic effectiveness of control measures selected is analyzed after the stand is harvested.

NPVUNPL is a stand level computer model which simulates growth loss in red pine plantations due to Saratoga spittlebug injury (Heyd, 1985). The program requires initial input of number of trees per acre and site index. Three rotation options are available in the model. First, harvest at age 45 with no thinning. Second, harvest at age 80 with thinning to a residual basal area level of

90 square feet per acre every 10 years after age 33. Third, harvest at age 80 with thinning to a residual basal area level of 120 square feet per acre every 10 years after age 33. Stumpage prices may be set for the products of each thinning as well as for the final harvest.

Insect control in NPVUNPL may be accomplished by chemically controlling alternate host vegetation, spraying chemical insecticides, or taking no action. Costs of stand establishment and insect control may be entered, or estimated by the program. Economic analysis of the management strategy employed is carried out using Net Present Value at interest rates between 7% and 10%. Analysis is performed for the same stand with three different Saratoga spittlebug hazard ratings of increasing severity (Heyd, 1985).

The objective of the current project was to develop a computerized growth and yield model (RPPEST), which allows simulation of the effects on red pine growth and yield of the major insect pests of red pine. This model is seen as a training tool for land managers, enabling them to experience the effects of insect attack and judge the soundness of alternative control practices.

Overview of the Model

RPPEST is a stand level model for plantations and natural stands which simulates the impact on red pine growth and yield of four insects: white grubs, pine root collar weevil, Saratoga spittlebug, and red-headed pine sawfly. Figure 1 is a flowchart of RPPEST. The model allows simulation of the impact of one of the insects, or a stand may be grown without insects. If the insect attack alternative is selected two stands are grown simultaneously, one with insect controls available in the model, and the other with no insect control measures. This allows an economic evaluation of the insect control strategy selected.

Initial stand information needed to run the model includes site index, minimum pole and sawtimber diameters, seedling age at planting, stand age at the beginning of simulation (0 if insect attack is selected), initial basal area per acre, trees per acre, and harvest age. Maximum stand age allowable at harvest is 135 years. If the insect attack option is selected, alternative planting and site preparation methods are available.

RPPEST includes the ability to establish a thinning regime for a stand. Thinning selects trees equally across the stand's diameter distribution. Thinning may be to a

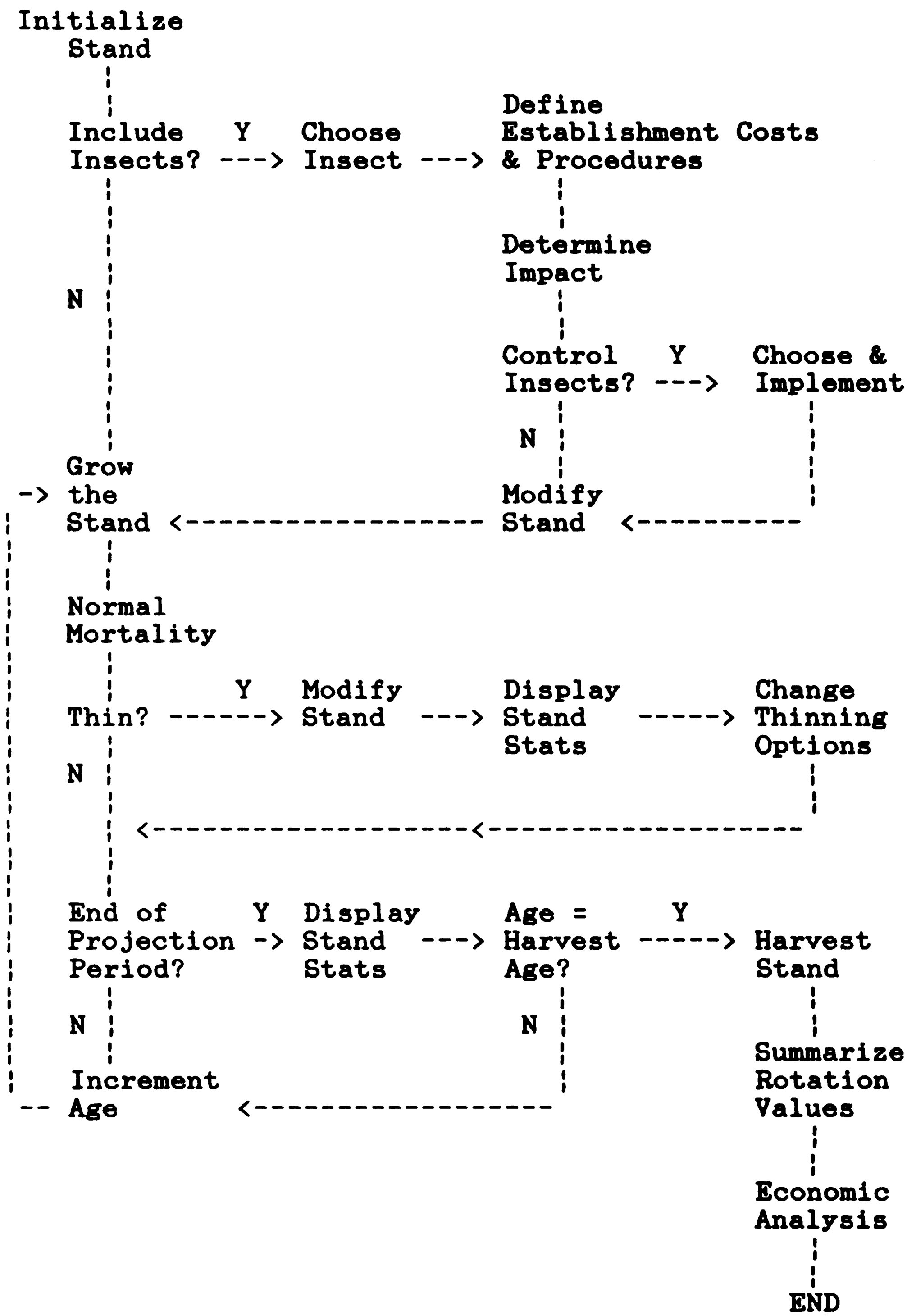

Figure 1. A Flowchart of RPPEST.

residual basal area, or may remove a specific number of trees per acre, or may remove a certain proportion of the stand's basal area or trees per acre. If insect attack is simulated both stands follow the same thinning regime.

The stand is grown until the end of the specified projection period or when a thinning year is reached. A stand summary is then displayed and the pattern is repeated until harvest age is reached. A final volume summary for each stand is displayed. Then an economic analysis of overall management and of wood saved by the insect control measures is displayed.

Plantation Establishment

There are many ways to prepare a potential red pine site for planting. These methods may be grouped into two broad categories: mechanical and chemical (Sajdak, 1982). These two categories, along with no site preparation, are available as options in setting up stand regeneration in the model.

Winebar and Gunter (1984) conducted a survey of forest management costs in 1982 among government agencies, forest industry and private contractors in Michigan, Wisconsin, and Minnesota. Costs for site preparation were taken from the results of that survey. Costs used in this model have been adjusted for inflation from 1983 to 1985, the last date for which figures are available, using the producer price index. The adjustment factor is 4.6%. Winebar and Gunter arrived at adjusted average figures of $ 64.57 per acre for mechanical preparation and $ 52.30 per acre for chemical preparation.

Red pine seedlings are planted either by hand or by machine. Both options are available. Planting stock is assumed to be bareroot. Costs for planting are categorized by the type of site preparation preceding the planting and by the number of seedlings per acre being planted (Winebar and Gunter,1984). Early survival of planted red pine can vary significantly depending upon several factors, such as quality of planting stock, degree of site preparation, date of planting (Emch and Carvell, 1985), use of correct planting techniques, and weather. Attempting to combine all of these elements into a model would be a major task in itself, so a simplifying assumption was made. It was assumed that things went reasonably well during the planting operation, except for some small degree of uncertainty. That is, seedlings were handled with reasonable care, seedlings were not J-rooted during planting, or a two month drought did not set in right after planting.

Published average first year red pine seedling survival data for the period 1934-1974 on the Huron-Manistee National Forest in the northern Lower Peninsula of Michigan was 83%, while 1982, 1983, and 1984 plantings showed first year survival rates of 85%, 94% and 84% respectively (Labumbard, 1984). These figures were used as the basis for assuming a first year survival rate of 84%. To account for year to year variation, the model randomly varies first year survival between 74% and 93% in any given year (84% + or - 9%). Published seedling survival data is uncommon, and the Huron-Manistee information covers a longer time span than other Lake States data found in the literature. The applicability of this data to areas outside northern Lower Michigan is unknown.

Type of planting (hand or machine) is assumed to have no effect on first year survival (Knighton, 1972). No data was available to show a difference between chemical and mechanical site preparation in the first year, so none was assumed. Carmean (1971) found first year survival of 2-0, 3-0, and 4-0 seedlings planted in Minnesota on sandy soil with grass competition to average 53.9% when no site preparation was used. If no site preparation is indicated in the model, first year survival is reduced by a randomly determined additional 0% to 30% (the difference between the mean value of 84% and the 53.9% observed by Carmean) to reflect the added uncertainty of possibly heavy vegetative competition on the seedlings.

Red Pine Growth and Yield

Accurate prediction of red pine growth and yield throughout the life of a stand is fundamental to this model. The equations used to accomplish this forecasting were drawn from the forestry literature. The general form of the growth and yield section of the model and many of the equations used, including those for basal area growth and tree volume, were drawn from a red pine growth and yield computer model called REDPINE, written and briefly described by Lundgren (1981).

Red Pine Growth

Red pine growth equations considered for use in the model have been published by Buckman (1962), Chen and Rose (1977), USDA Forest Service (1979), Lundgren (1981), and Reed et al. (1986).

Chen and Rose (1977) developed an individual tree basal area growth model for red pine using data from a plantation in Wisconsin. Their model estimates a tree's basal area growth as a function of the tree's basal area and competition with its neighbors. This model does not seem to adequately explain basal area growth in thinned stands, and the calculations are difficult. Therefore, this model was not used in RPPEST.

Staff members of the USDA Forest Service's North Central Experiment Station have developed an individual tree growth and yield projection system for the Lake States called STEMS (USDA Forest Service, 1979). Red pine is among the species included in this diameter growth model. The data used to deveolp this model were collected in Michigan, Wisconsin, and Minnesota. The STEMS model requires information about each tree in the stand to be modeled, including crown ratio and diameter. It was felt that these extensive information requirements would be a burden to people using RPPEST, so the STEMS diameter growth model was not used in RPPEST.

Reed et al. (1986) developed a stand level metric cubic volume and basal area estimation system from data collected in red pine plantations in both the Upper and Lower Peninsulas of Michigan. The basal area equation requires only stand age, basal area, and site index at a beginning age to estimate basal area at a future age. Reed et al. suggest that this stand model not be used in thinned stands. The ability to simulate thinning was a fundamental goal in the development of RPPEST, and so precluded this model's use.

Equations from Buckman (1962) and Lundgren (1981) were selected to predict red pine basal area growth in RPPEST. From the time when breast-height age is greater than zero until total tree age equals 24, stand basal area per acre is estimated by an equation developed by Lundgren from data originally collected by Wambach (1967):

Basal area = 6.565 * site index * ((1 - e(-0.04018 * total age))^1.1677) * (1 - e(-0.01885 * trees/ac.))

Wambach's data were collected in unthinned Lake States plantations up to 35 years old that were established at various initial spacings. Lundgren's model is used rather than Wambach's model because it explains 10% more of the natural variation present in the data, and has 5 square feet of basal area per acre less error associated with it's predictions. The first derivative of Lundgren's equation is used to compute annual basal area growth per acre as follows:

Basal area growth = 6.5653 * site index * (A1 * A2 - A3 * A4)

A1 = e(1.1677 * LN(1 - e(-0.040172 * (total age + 1))))

A2 = 1 - e(-0.0018854 * trees/ac.)

A3 = e(1.1677 * LN(1 - e(-0.040172 * total age)))

A4 = 1 - e(-0.0018854 * trees/ac.)

Buckman (1962) developed a stand level red pine basal area growth equation from data collected in Minnesota.

Most of the plots measured were located in thinning studies in what appear to be both natural stands and plantations. Site index of the experimental stands varied from 46 to 63 feet, while stand age ranged between 21 and and 144 years. Annual basal area growth per acre for stands where tree age is greater than 24 years old is computed using Buckman's equation:

$$\text{Basal area growth} = 1.689 + 0.04107 * \text{basal area/ac.} - 0.000163 * (\text{basal area/ac.})^2 - 0.07696 * \text{total age} + 0.0002274 * \text{total age}^2 + 0.06441 * \text{site index}$$

The two basal area growth equations used in RPPEST were also used in Lundgren's REDPINE model (1981). Lundgren (1983) compared total peeled cubic foot volume per acre mean annual increment (MAI) predicted by REDPINE with red pine MAIs from Michigan, Minnesota, Wisconsin and Ontario reported in the literature. A simple regression equation relating REDPINE predicted MAI to observed MAI had a coefficient of determination of .73, and an F test showed that the equation was not significantly different (at the 0.05 level) from the assumption that predicted MAI equaled observed MAI. The accuracy of prediction and the apparently wide area of geographic applicability were determining factors in the selection of Buckman's and Lundgren's basal area growth models for use in RPPEST.

Breast-height age is computed using an equation from Benzie (1977) developed by A. L. Lundgren from data collected by Wambach (1967):

$$\text{Breast height age} = \text{total age} - 10.5 + 0.05 * \text{site index}$$

Lundgren (1981) developed a maximum annual diameter increment function to limit the growth of stands with low numbers of trees per acre, or young thinned stands. Occasionally the diameter growth predicted for these types of stands appears to be too high. To remedy this situation, the following constraint is used in RPPEST:

$$\text{Maximum dbh increment} = .007 * \text{site index} * e(-0.01 * \text{breast-height age})$$

Using the maximum diameter increment calculated as above, maximum annual basal area growth increment is:

$$\text{Maximum BA growth} = 0.00545415 * \text{trees/ac.} * (2 * \text{mean dbh} * \text{maximum dbh increment} + \text{maximum diameter increment}^2)$$

Mean dbh, or quadratic mean diameter, is the diameter of the tree of average basal area.

Diameter Distribution

Hafley and Schreuder (1970) discuss several statistical distributions used for fitting diameter class data for even-aged stands, including the beta, Johnson's S_B, lognormal, gamma, Weibull, and normal. The only published information for red pine concerning any of these distributions is found in Zarnoch et al. (1982), who fit diameter class data for various thinning regimes in red pine to the Weibull distribution. Weibull parameters are given for stands of age 25, 32, and 39 years. Because Zarnoch et al. have no data for stands older than 39 years it was decided not to use this information in a model where maximum stand age can be 135 years.

Lacking other information, and after examining graphs of red pine diameter distributions, it was decided that red pine diameter distributions would be approximated by the normal distribution. Before a given number of trees per acre can be partitioned into one inch diameter classes using the normal distribution, an estimate of the standard deviation of the diameters must be calculated. A function of this type was developed by Lundgren (1981) using data published by Stiell and Berry (1973):

$$\text{Standard deviation of dbh} = 0.37628 * \text{mean diameter} * e(-0.093346 * \text{mean diameter})$$

There were two reasons for assigning stand level data to one inch diameter classes. First, the mortality function selected requires dbh to calculate survival rates. Second was the need to partition basal area into merchantability classes for product specification. No other use was made of the diameter class information.

Estimating Red Pine Volume

Tree height is one component of the volume equations used in the model. An equation to estimate total height of dominant and codominant trees was developed by Lundgren and Dolid (1970) to describe published site index curves for red pine in the Lake States:

$$\text{Total height} = 1.89 * \text{site index} * (1 - e(-0.01979 * \text{age})\hat{\ }1.3892)$$

Several equations are used in the model to calculate red pine volume per acre. Total cubic foot volume per acre, utilizing the entire stem minus bark, is calculated using an equation from Buckman (1962):

$$\text{Cubic feet/ac.} = 0.4085 * \text{basal area/ac.} * \text{average total stand height}$$

Cordwood volume per acre, including bark, to a three inch minimum top diameter inside bark is calculated using an equation developed by Buckman (1962):

Cords/ac. = 0.003958 * basal area/ac. * average total stand height

Merchantable cubic foot volume per acre to a three inch minimum top diameter inside bark is computed using a formula derived by multiplying the above cordwood volume equation from Buckman (1962) by 79 cubic feet per cord (Lundgren, 1981):

Cubic feet/ac. = 0.3127 * basal area/ac. * average total stand height

International 1/4 inch board foot volume per acre to a six inch top is calculated using a ratio of board feet per total cubic foot (Lundgren, 1981) in sawtimber size trees:

Board feet/cubic foot = -8.76 + 1.985 * dbh - 0.07253 * dbh^2 + 0.0008421 * dbh^3 + 0.04951 * average total stand height - 0.00892 * dbh * average total stand height + 0.0003169 * dbh^2 * average total stand height - 0.000002786 * dbh^3 * average total stand height

Estimating Red Pine Mortality

Buchman (1983) has developed a model that predicts survival for several Lake States tree species, including red pine. The data used to develop the model has strength in both number of samples and geographic extent. Equation coefficients for red pine were developed from nearly 40,000 one year tree records gathered from throughout the Lake States.

Performance of this tree survival model was tested against USDA Forest Service permanent plot data. For red pine between 1 inch and 16 inches dbh the aggregate predicted survival rate was only 0.3% higher than the measured survival rate (Buchman, 1985). This model was therefore selected to estimate annual red pine survival in trees 1 inch dbh and larger:

Survival rate = 0.9997 - (1 / (1 + e^n))

n = 1.9953 + 57.97 * dbh growth rate^1.012 + 0.2648 * (dbh - 1)^1.626 * e(-0.1273 * (dbh - 1)).

Region of Applicability for Insect Models

Much of the entomological information used in RPPEST was collected by Louis F. Wilson of the USDA Forest Service North Central Experiment Station in East Lansing, Michigan, and his students R.D. Averill and R.L. Heyd. Most of their information was collected in the northern Lower and Upper Peninsulas of Michigan. Dr. Wilson's job is to perform entomological research applicable to the north central United States. Thus, the data used in RPPEST is pertinent throughout the Lake States (G.A. Simmons, 1987, Michigan State University, personal communication).

Wide applicability of information about a pest does not mean that the insect is a significant problem everywhere it is found. Impact may vary as climate and vegetation change across a region. For example, potential impact of pine root collar weevil in northern Lower Michigan declines as the climate becomes harsher moving from west to east across the state (Wilson and Millers, 1983). Therefore, utility of particular sections of this model may vary throughout the region where it is potentially applicable.

White Grubs

White grubs (Coleoptera, Scarabaeidae) are the larvae of May beetles and other related beetles. The genus Phyllophaga contains some of the largest and most destructive larvae. They occur in the soil and feed on the roots of trees, grasses, and other vegetation. The larvae consume the smaller roots and girdle the larger roots of young red pine seedlings. The damage caused by this feeding manifests itself on the trees in two ways: reduced growth, and mortality (Fowler and Wilson, 1971).

White Grub-Caused Mortality

Fowler and Wilson (1971, 1974) studied red pine seedling mortality caused by white grubs in the Upper Peninsula of Michigan from the time of planting through the fifth growing season. Most white grub caused mortality took place in the first three years after planting (Fowler and Wilson, 1974). The data in Table 1 were interpolated from Figure 6 in Fowler and Wilson (1971). The graph shows a triangular region of seedling mortality described by lower and upper limits of red pine seedling mortality through the second growing season for a given mean number of Phyllophaga larvae per cubic foot of soil. Because non-insect-caused mortality is accounted for in RPPEST, a base rate of 4% non-grub mortality was subtracted from each value interpolated from the graph.

Table 1. Lower and upper limits of red pine seedling mortality through year 2, for a given mean number of Phyllophaga per cubic foot of soil.

Mean # grubs per cubic ft.	% Mortality lower limit	% Mortality upper limit
0.00	0	0
0.25	6	14
0.50	12	30
0.75	18	44

In order to duplicate the lines representing the lower and upper limits of seedling mortality in the graph, linear regression was used to fit simple equations to both sets of interpolated data using number of grubs per cubic foot of soil as the independent variable. Thus, the R^2 values shown below should only be used to judge the precision of the reproduction of the graph lines. The resulting equations are shown below:

% Seedling mortality lower limit = 24 * # of grubs

R^2 = 1

% Seedling mortality upper limit = 58.85714 * # of grubs

R^2 = .9997

Simulation of first and second year mortality of red pine seedlings begins with generation of a uniformly distributed random number between 0 and 0.75. This represents the mean number of grubs per cubic foot of soil. This number is then used in the regression equations to determine the lower and upper levels of mortality possible for this particular grub population. First and second year percent mortality is then determined by generating a random number between the calculated lower and upper bounds.

Third and fourth year seedling mortality were interpolated from Figure 2 in Fowler and Wilson (1974). Third year mortalities for four plantations were 1%, 3%, 4%, and 12%. Third year mortality is assumed to vary between 1% and 4% with a probability of .75, and between 4% and 12% with a probability of .25.

Fourth year mortality from Fowler and Wilson (1974) is essentially 0% for three plantations and 3% for the fourth. Fourth year mortality is assumed to be 0% with a probability of .75, and to vary between 1% and 3% with a probability of .25.

Mortality percentages from the first four years are summed and number of trees per acre is reduced by this amount.

Growth Loss Due To White Grubs

Fowler et al. (1982), in an extension of the work by Fowler and Wilson (1971, 1974), reported that differences in red pine site index values between plots treated with an insecticide effective on white grubs and plots that received no treatment were 5, 7, 8, and 9 units among four plantations. This study was conducted in the Upper Peninsula of Michigan on plantations which had completed ten growing seasons.

To simulate the effect of grubs on height growth a uniformly distributed random number between 5 and 9 is generated, and site index is reduced by this amount.

Control Measures and Efficacy

Control of white grubs is accomplished by applying a pesticide such as aldrin at the time of planting an old field site (Fowler and Wilson, 1971). Fowler and Wilson (1974) report that application of aldrin reduced white grub caused red pine seedling mortality by 83% in the first five years. If white grub control is chosen in the model, the combined four year mortality figure is reduced by 83%. Also, site index is assumed to return to normal.

Pine Root Collar Weevil

The pine root collar weevil (*Hylobius radicis* Buch.) causes mortality and growth loss in red pine. This weevil is found throughout northeastern North America. Trees between the ages of 8 and 15 years are most vulnerable to attack (Wilson and Kennedy, 1970). The injury is caused by larval feeding in the inner bark of the root collar. A mean number of 2.2 immature insects per tree is considered the threshold of tree mortality (Wilson and Millers, 1983). As a red pine plantation approaches crown closure the environment around the base of the tree becomes less favorable for weevils and populations begin to decline naturally (Wilson and Millers, 1983).

Wilson and Kennedy (1970) described three pine root collar weevil hazard zones in Northern Lower Michigan. The zones are categorized by their degree of potential weevil damage. Wilson also quantified the probability of heavy damage (death) to red pine in a hazard zone relative to the distance to the nearest weevil infested stand. Table 2 is reproduced from Wilson and Kennedy (1970).

Table 2. Probability of heavy damage to red pine from root collar weevil, by hazard zone and distance to a weevil infestation (Wilson and Kennedy, 1970).

Distance to weevil infestation	Hazard Zones		
	Low	Medium	High
Miles	%	%	%
0 - 1/8	5	15	>50
1/8 - 1/2	5	10	40
1/2 - 1	5	5	25
1+	<5	5	10

Stand damage is divided into three categories: light, moderate, and heavy. Red pine mortality in lightly damaged stands will be less than 5%. Damage is usually light in the low hazard zone. Red pine mortality in moderately damaged stands will range from 5% to 10%. In the medium hazard zone, damage can be light or moderate. Heavily damaged stands will have red pine mortality of 10% to 15%. In the high hazard zone, damage is usually moderate or heavy (Wilson and Millers, 1983).

Pine root collar weevil reduces red pine height growth beginning two growing seasons before the tree dies. Two years before death height growth is reduced about 7%. The year before death height growth is reduced about 24% (Schmiege, 1958). The model assumes that trees which experience significant growth loss are very close to death and will probably die within a short time. Rather than accounting for a 24% loss of height growth the year before an attacked tree dies, growth loss is ignored and the tree is assumed to die.

There are two methods of controlling poulations of pine root collar weevil. The first is spraying insecticide such as lindane around the base of each tree (Finnegan and Stewart, 1962). One application between ages 6 years and 10 years will usually keep the trees growing well until the crowns begin to close and weevil populations begin to decline naturally (Wilson and Millers, 1983).

The second method involves pruning the branches of the lower 2 feet of the tree trunk and scraping the litter and topsoil from around the base of the tree for a distance of 1 foot. This treatment usually suppresses the weevil population for 4 or 5 years and allows the tree to reach crown closure safely (Wilson, 1967).

Simulating Pine Root Collar Weevil Impact on Red Pine

Simulation begins by randomly assigning the stand to a weevil hazard zone, and setting a random distance to the nearest weevil infested stand. The probability of heavy damage is determined using the hazard zone and distance to infestation (see Table 2). A random number between 0 and 1 is now generated. If this number is less than the probablility of heavy damage then percent red pine mortality is determined based on the hazard zone. If the random number is greater than the probablility of heavy damage no red pine mortality occurs.

If the hazard zone is low then mortality percentages between 0% and 5% are possible. In the medium hazard zone mortality can range from 0% to 10%. Red pine mortality in the high hazard zone can take on values from 5% to 15%. The number of trees per acre is then reduced equally across all diameter classes to account for the simulated mortality. The assumption is made that the trees which experience growth loss will die, and they are included in the mortality percentage generated above.

Both control options outlined above are possible in the model. Control takes place in year 8 after planting. Adult weevil surveys are automatically scheduled for year 4 and year 7 after planting if this option is chosen, and their cost is included in the economic analyses.

Saratoga Spittlebug

The Saratoga spittlebug (Aphrophora saratogensis [Fitch]), found throughout the geographic range of red pine, is responsible for mortality, deformity, and loss of growth in red pine plantations where trees are between 3 and 15 feet tall. Adult spittlebugs feed by inserting their beak into red pine twigs of the previous year's growth and sucking plant juices out. The puncture wound and the bug's saliva create necrotic tissue and stimulate resin accumulation. This blocks fluid transport in the xylem and phloem and can eventually kill the branch. If feeding is heavy enough, the branch will flag. The bugs prefer to feed in the upper whorl of branches. If the top branches are killed, lower whorls compete for terminal dominance, resulting in a height growth loss. Crooked or forked trees usually are the result (Heyd, 1978). If enough feeding occurs the tree will die.

In order to complete its life cycle the immature spittlebugs require that one of several alternate host plants be available for feeding. The primary alternate host is sweet-fern, Comptonia peregrina Coult. However there are approximately 200 other secondary hosts including blackberry, raspberry, bracken fern, blueberry, goldenrod, and Sumac (Wilson, 1971). Without any of these alternate hosts, the Saratoga spittlebug cannnot complete its development.

Wilson (1971) developed a method to rate the risk of Saratoga spittlebug injury to a potential planting site, or an existing stand. The technique involves estimating the percent ground cover of sweet-fern and other alternate host plants. Wilson published a risk rating graph which uses the two cover values to place a stand into one of three risk categories: low, moderate, or heavy.

Heyd (1978) studied the impact of Saratoga spittlebug on red pine in northen Lower Michigan and described the growth loss as uniformly affecting the diameter and height growth of the tree. A method was developed by Heyd which ties growth loss to Wilson's plantation risk categories. Red pine on low risk sites can lose a maximum of 4 years growth. Trees on moderate risk sites can lose between 4 and 10 years of growth, depending upon spittlebug population levels. Trees on heavy risk sites can lose their entire volume to mortality and loss of merchantability due to crooked and misshapen stems.

There are two ways to control the impact of the Saratoga spittlebug on red pine growth and yield. The first is by using an insecticide (such as malathion) to directly reduce the spittlebug population. Wilson and Millers (1966) found that malathion reduced Saratoga spittlebug populations by 99% in northern Lower Michigan. Many insecticides, including malathion, are not selective for spittlebug and may kill beneficial predatory insects.

The second control method involves reducing the population of alternate host plants, especially sweet-fern. Depending upon the size of the plantation this control may be either mechanical or chemical. If the area is small, mowing could be sufficient. However, mowing encourages resprouting of sweet-fern and actually leads to increased density (Heyd et al., 1987).

Chemical control seems to be much more promising, if more expensive. Heyd et al. (1987), working in northern Lower Michigan, found that 2 or 3 quarts of Roundup per acre will effectively control sweet-fern, and other alternate hosts such as blueberry, at least into the fourth year after treatment. Three years after treatment sweet-fern ground cover was only 4%, down from a pre-treatment level of 31%. An added benefit of chemically controlling alternate hosts is the reduction in vegetative competition with the young red pine. This may increase red pine site index by 5 units on moderate risk areas if site index is between 50 and 70, and by 10 units on high risk sites for the same range of site indexes (Heyd, 1982).

Simulating Saratoga Spittlebug Impact on Red Pine

Simulation of Saratoga spittlebug impact begins with the random generation of sweet-fern percent ground cover. The ground area not covered by sweet-fern is randomly assigned a percent cover for all other alternate host plants. These two numbers are used to determine the spittlebug risk rating of the plantation. The next step is determination of the extent of damage to the stand.

If the risk rating is low, then a random number between 0 and 4 is generated to represent the number of years of growth that will be lost. If the risk category is moderate, then a random number between 4 and 10 is generated to represent years of lost growth. If the risk category is high then the stand is assumed to have a 50% chance of losing between 10 years and 20 years of growth, and a 50% chance of losing all merchantable volume. So a random number is generated to represent the probability and growth loss is assigned according to its value.

Alternative control strategies are available in the model to offset red pine growth loss. Selection of chemical control of alternate hosts at the time of planting is assumed to keep the stand in the low risk category until it reaches crown closure. At crown closure the alternate host plants are shaded out by the trees and the risk from Saratoga spittlebug is removed. Site index is also increased if herbicide is applied (Heyd, 1982).

Selection of insect control using insecticide, following Heyd (1985), is assumed to require one application for low and moderate risk sites, and two sprays for a high risk site. If site index is less than 50 then a second spray is needed on moderate risk sites and a third spray is needed on high risk sites to compensate for the extended period of time the trees are vulnerable due to slow height growth (Heyd, 1981). The assumption is made that the insecticide will adequately protect the plantation.

Red-headed Pine Sawfly

The red-headed pine sawfly (<u>*Neodiprion lecontei*</u> [Fitch]) causes mortality, deformity, and growth loss in red pine 3 feet to 15 feet tall. The tree is injured as the sawfly larvae consume its needles. Over the past 50 years the red-headed pine sawfly has been the third most important target for insect suppression programs on National Forests in the eastern portion of the U.S (Averill et al., 1982).

Susceptibility of red pine to sawfly defoliation is directly related to tree vigor. Water stress is a very important factor in the tree's ability to resist sawfly attack. Extended periods of drought or poor quality sites can predispose a stand to attack (Averill et al., 1982).

Averill (1977) and Averill et al. (1982), in studies conducted in northern Lower and Upper Michigan, Wisconsin, New York, and Ontraio, Canada, proposed a method of classifying plantation sites into Site Resistance Classes (SRC) based on ground cover, distance to hardwood edges, and soil characteristics. Trees on SRC I sites are very resistant to sawfly attack, and sawfly populations are widely scattered. Trees on SRC II sites are fairly resistant, and sawfly populations are scattered with some trees moderately infested. SRC III sites are very dry, sawfly populations are high, and trees are generally heavily infested. Site index on SRC III sites is usually near or below 50 (Averill et al., 1982). In Northern Lower Michigan alone there are over 92,000 acres of red pine on sites where the site index is less than 50 (Jakes, 1982).

Sawfly damage on SRC I sites is slight, and tree mortality is rare. Two SRC I sites had mortality of 1.3% and 0% after a three year red-headed pine sawfly outbreak. Pulpwood yield was reduced very slightly, and sawlog yield was not affected (Averill et al., 1982).

Red pine on SRC II sites also experience very little problem. A few trees are killed (1.8% in the outbreak mentioned above) and others are top killed, resulting in forked or crooked trees. Again, pulpwood yield was slightly affected, and sawlog yield was not affected (Averill et al., 1982).

Red-headed pine sawfly causes serious problems for red pine on SRC III sites. Tree mortality is rapid and wide spread (52.4% in the outbreak mentioned above), and trees that remain alive are stunted and severly deformed. Merchantable volume is usually less than 1% of the volume remaining (Averill et al., 1982).

There are two ways of controlling the red-headed pine sawfly: chemical and cultural. The chemical control consists of applying insecticide to heavily infested SRC III areas in the early stages of the population build-up. Cultural control involves suppressing vegetation which competes with young red pine for moisture and other resources needed for growth.

Simulating Red-headed Pine Sawfly Impact on Red Pine Volume

Simulation of red-headed pine sawfly impact begins with the generation of a random number between 6 and 13, which represents the simulated stand's age when the sawfly population reaches outbreak levels. Most tree mortality occurs in one year (Averill et al., 1982). Red pine stands are not often composed entirely of one site resistance class, so the next step is to determine the proportion of the stand in each SRC. These proportions are randomly set. If site index is above 50, there will be no SRC III areas in the plantation.

Percent mortality for the trees in the SRC I portion of the plantation is randomly set between 0 and 1.3%. Mortality percent for SRC II trees is randomly set between 0 and 1.8%. All trees in the SRC III areas of the plantation are assumed to be dead or unmerchantable, and are removed from the trees per acre count.

Both control options mentioned above, application of insecticide and application of herbicide at time of planting, are available in the model. Sawfly populations remain at relatively constant levels until the year of a population outbreak, when they rise dramatically (G.A. Simmons, 1987, Michigan State University, personal communication). For this reason, insecticide application is assumed to occur in the randomly generated outbreak year (G.A. Simmons, 1987, Michigan State University, personal communication), and be effective enough to reduce tree mortality to SRC II levels. Herbicide application is also assumed to be effective in maintaining tree vigor to the point where SRC II mortality levels are achieved.

Economic Evaluation

To compare costs and revenues which occur throughout a lengthy rotation on an even basis, the values must be discounted to a common year. In this model that is the year of planting, year 0. Gunter and Haney (1984) suggest that for ease of calculation all costs and revenues should be considered to occur at the end of the year. This convention is followed in RPPEST.

Two analyses are performed after Drapek (1985). The first compares all costs with all revenues to evaluate the overall management strategy using Net Present Value (NPV). Net Present Value is calculated by subtracting the sum of the discounted costs from the sum of the discounted revenues. If the result is positive then the red pine stand in question compares favorably with alternative investments with real rates of return of 4% or less.

The second analysis also uses NPV to evaluate insect management efforts. The discounted value of the wood saved by monitor and control techniques is compared with the discounted costs of those techniques. Again, a positive NPV indicates a worthwhile investment.

Summary

RPPEST is a stand level model designed as a training tool for land managers which simulates the impact of four major insect pests on red pine. The model allows a wide range of initial site conditions and the establishment of thining regimes. Insect control measures are available in the model, and an economic analysis using Net Present Value is performed after the stand is harvested.

The growth and yield portion of the model has been shown to be applicable throughout the Lake States and southern Ontario (Lundgren, 1983). The insect impact models used in RPPEST should also apply throughout the same region. However, these models should be tested for accuracy on a local basis.

Additional information is needed regarding early survival of planted red pine, especially beyond the first year and in relation to site and climatological factors. Once the tree has reached about five feet in height, mortality seems to be fairly well quantified. More information is also needed concerning red pine diameter distributions. The Weibull distribution has the flexibility to accurately describe diameter distributions, and an extension of the work of Zarnoch et al. (1982) would be a valuable addition to the literature.

Much remains to be learned about the insects included in this model. Factors involved in population dynamics of pine root collar weevil, Saratoga spittlebug, and red-headed pine sawfly are not well documented. The length of population cycles and the factors which combine to promote insect outbreaks need further study. Information about mortality rates and their causes at each stage of the insects' life cycles are not known.

Additional investigation is also required into the effects of Saratoga spittlebug and white grub damage as previously injured trees age. Quantification of the relationship between site quality and insect damage would be directly beneficial to land managers trying to assess the impact of an insect outbreak just getting under way.

Information about interaction between these insects does not exist. For example, does white grub injury predispose a stand to attack, or increased levels of damage, by pine root collar weevil?

The information needs outlined above present a formidable challenge to entomologists and forest land managers. However, the addition of this information to what is already known about the interactions between the forest and these insects would greatly assist forest and forest insect management in the Lake States.

LIST OF REFERENCES

LIST OF REFERENCES

Averill, R.D. 1977. Impact of red-headed pine sawfly, *Neodiprion lecontei*, on young red pine plantations. Ph.D. Dissertation, 127 p. Michigan State University, E. Lansing.

Averill, R.D., L.F. Wilson, and R.F. Fowler. 1982. Impact of the redheaded pine sawfly (Hymenoptera: Diprionidae) on young red pine plantations. Great Lakes Entomol. 15(2): 65-91.

Belcher, D.M. 1982. TWIGS: The woodsman's ideal growth projection system. P. 70-95 in Microcomputers, a new tool for foresters. Purdue Univ. Dept. For. & Nat. Res. West LaFayette, IN.

Belcher, D.M., M.R. Holdaway, and G.J. Brand. 1982. A description of STEMS, the Stand and Tree Evaluation and Modeling System. USDA Forest Service Gen. Tech. Rep. NC-79, 18 p.

Benzie, J.W. 1977. Manager's handbook for red pine in the North Central states. USDA Forest Service Gen. Tech. Rep. NC-33, 22 p.

Benzie, J.W. and J.E. McCumber. 1983. Red pine. P. 89-91 in R.M. Burns (tech. compiler). Silvicultural Systems for the Major Forest Types of the United States. USDA Forest Service Ag. Hndbk. 445, 191 p.

Brand, G.J. 1979. Timber management guides and marking rules. P. 56-60 in A Generalized Forest Growth Projection System Applied to the Lake States Region. USDA Forest Service Gen. Tech. Rep. NC-49, 96 p.

Buchman, R.G. 1979. Mortality functions. P. 47-55 in A Generalized Forest Growth Projection System Applied to the Lake States Region. USDA Forest Service Gen. Tech. Rep. NC-49, 96 p.

Buchman, R.G. 1983. Survival predictions for major Lake States tree species. USDA Forest Service Res. Paper NC-233, 7 p.

Buchman, R.G. 1985. Performance of a tree survival model on national forests. N. J. Appl. For. 2(4): 114-116.

Buckman, R.E. 1962. Growth and yield of red pine in Minnesota. USDA Forest Service Tech. Bull. 1272, 50 p.

Carmean, W.H. 1971. Large, well balanced stock and control of grass competition needed for red pine plantings on sandy soils. Tree Planters' Notes. 22(1): 8-10.

Chen, C.M. and D.W. Rose. 1978. Nonlinear basal area growth models for red pine. P. 17-20 in Proceedings 1977 Midwest Forest Mensurationists Meeting. USDA Forest Service Gen. Tech. Rep. NC-46.

Drapek, R.J. 1985. A simulation of jack pine budworm and white pine weevil monitor and control techniques and their effects on jack pine economics. M.S. Thesis, 118 p. Michigan State University, E. Lansing.

Emch, D.J. and K.L. Carvell. 1985. Effects of cold storage and planting date on red pine survival and growth. N. J. Appl. For. 2(2): 51-53.

Finnegan, R.J. and K.E. Stewart. 1962. Control of the pine root collar weevil, Hylobius radicis. J. Econ. Entomol. 55: 482-486.

Fowler, R.F., and L.F. Wilson. 1971. White grub populations, Phyllophaga spp., in relation to damaged red pine seedlings in Michigan and Wisconsin plantations (Coleoptera: Scarabaeidae). Michigan Entomol. 4(1): 23-28.

Fowler, R.F.and L.F. Wilson. 1974. Injury to Aldrin-treated and untreated red pine by white grubs (Coleoptera: Scarabaeidae) and other agents during first five years after planting. Michigan Entomol. 7(3): 81-88.

Fowler, R.F., L.F. Wilson, and A.L. Lundgren. 1982. Projected red pine yields from Aldrin-treated and untreated stands damaged aby white grub (Coleoptera: Scarabaeidae) and other agents at stand age ten years. Great Lakes Entomol. 15(3): 165-170.

Gunter, J.E. and H.L. Haney. 1984. Essentials of Forestry Investment Analysis. 337 p. OSU Bookstores, Corvallis, OR.

Hafley, W.L. and H.T. Schreuder. 1977. Statistical distributions for fitting diameter and height data in even-aged stands. Can. J. For. Res. 7: 481-487.

Heyd, R.L. 1978. An evaluation of the impact of the Saratoga spittlebug, Aphrophora saratogensis (Fitch), on the growth of red pine, Pinus resinosa Ait. Ph.D. Dissertation, 80 p. Michigan State University, E. Lansing.

Heyd, R.L. 1982. Insects versus artificial regeneration of conifers. P. 358-374 in G.D. Mroz and J.F. Berner(compilers). Proc. Artif. Regen. of Conifers in the Upper Great Lakes Region, 26-28 Oct. 1982, Green Bay, WI, 435 p. Mich. Tech. Univ., Houghton.

Heyd, R.L. 1985. NPVUNPL. A Saratoga spittlebug simulation program. Available from the Michigan Cooperative Forest Pest Management Program. East Lansing, Michigan.

Heyd, R.L. and L.F. Wilson. 1981. Risk-rating red pine plantations to predict losses from Saratoga spittlebug for management decisions. P. 93-98 in R.L. Hedden, S.J. Barras, and J.E. Coster (tech. coords.). Hazard-Rating Systems in For. Insect Pest Mgt.: Symp. Proc., 31 July-1 Aug. 1980, Athens, GA. USDA Forest Service Gen. TEch. Rep. WO-27, 169 p.

Heyd, R.L., R.L. Murray, and L.F. Wilson. 1987. Managing Saratoga spittlebug in pine plantations by suppressing sweetfern. N. J. Appl. For. 4(1): 16-17.

Jakes, P.J. 1982. Timber resources of Michigan's Northern Lower Peninsula, 1980. USDA Forest Service Res. Bull. NC-62, 120 p.

Knighton, M.D. 1972. Effect of ground preparation on planted red pine in Southwestern Wisconsin. USDA Forest Service Res. Note NC-134, 2 p.

Labumbard, H. 1984. Considerations in maintaining a quality planting program. P. 136-145 in R. Marty(editor). Managing Red Pine: Proceedings of the SAF Region V Technical Conference, 1-3 Oct. 1984, Marquette, MI, 273 p. Michigan State Univ., E. Lansing.

Lundgren, A.L. 1981. The effect of initial number of trees per acre and thinning densities on timber yields from red pine plantations in the Lake States. USDA Forest Service Res. Paper NC-193, 25 p.

Lundgren, A.L. 1983. New site productivity estimates for red pine in the Lake States. J. Forestry 81(11): 714-717.

Lundgren, A.L. and W.A. Dolid. 1970. Biological growth functions describe published site index curves for Lake States timber species. USDA Forest Service Res. Paper NC-36, 9 p.

Norris, F.W. 1987. Timber Mart North, 1st quarter 1987. Highlands, NC.

Ramm, C.W. and C.L. Miner. 1986. Growth and yield programs used on microcomputers in the north central region. N. J. Appl. For. 3(2): 44-46.

Reed, D.D., E.A. Jones, T.R. Bottenfield, and C.C. Trettin. 1986. Compatible cubic volume and basal area equations for red pine plantations. Can. J. For. Res. 16: 416-419.

Sajdak, R.L. 1982. Site preparation in the Upper Great Lakes Region. P. 209-214 in G.D. Mroz and J.F. Berner(compilers). Proc. Artif. Regen. of Conifers in the Upper Great Lakes Region, 26-28 Oct. 1982, Green Bay WI, 435 p. Mich. Tech. Univ., Houghton.

Schmiege, D.C. 1958. A study of the pine root collar weevil with particular reference to survey methods. M.S. Thesis, 85 p. Univ. Minnesota, Minneapolis.

Stage, A.R. 1973. Prognosis model for stand development. USDA Forest Service Res. Pap. INT-137, 32 p.

Stiell, W.M. and A.B. Berry. 1973. Yield of unthinned red pine plantations at the Petawawa Experiment Station. Canadian Forestry Service, Dept. of Envir., Publication 1320, 16 p.

USDA Forest Service. 1979. A generalized forest growth projection system applied to the Lake States region. USDA Forest Service Gen. Tech. Rep. NC-49, 96 p.

Wambach, R.F. 1967. A silvicultural and economic appraisal of initial spacing in red pine. Ph.D. Dissertation, 282 p. University of Minnesota, Minneapolis.

Wilson, L.F. 1971. Risk-rating Saratoga spittlebug damage by abundance of alternate-host plants. USDA Forest Service Res. Note NC-110, 4 p.

Wilson, L.F. and P.C. Kennedy. 1970. Pine root collar weevil hazard zones for red pine in Lower Michigan. USDA Forest Service Res. Note NC-104, 2 p.

Wilson, L.F. and I. Millers. 1966. Suppression of Saratoga spittlebug with helicopter application of low- and high-volume malathion. J. Econ. Entom. 59(6): 1456-1458.

Wilson, L.F. and I. Millers. 1983. Pine root collar weevil - its ecology and management. USDA Forest Service Tech. Bull. 1675, 33 p.

Winebar, B.L. and J.E. Gunter. 1984. Costs of forest management practices in the Lake States - including a comparison of costs between the major types of forestry organizations. P. 43-70 in R. Marty(editor). Managing Red Pine: Proceedings of the SAF Region V Technical Conference, 1-3 Oct. 1984, Marquette, MI, 273 p. Michigan State Univ., E. Lansing.

Zarnoch, S.J., C.W. Ramm, V.J. Rudolph, and M.W. Day. 1982. The effects of red pine thinning regimes on diameter distribution fitted to the Weibull function. Mich. State Univ. Ag. Exp. Sta. Res. Rep. 423, 11 p.

APPENDIX A

APPENDIX A

A User's Guide For RPPEST

Introduction

RPPEST is a computer program written in Turbo Pascal for IBM PC compatible microcomputers. The purpose of the program is to simulate the effects of insect damage on red pine growth and volume yield and to economically evaluate various control methods. Four insects are included in the program: white grubs, pine root collar weevil, Saratoga spittlebug, red-headed pine sawfly. The effect of one insect may be simulated, or a stand may be grown with no insects present.

When used only for red pine growth and yield, a stand may be joined at any point in it's life up to the maximum rotation age of 135 years. Initial number of trees per acre, initial basal area per acre, merchantability limits, and thinning regimes are all flexible within certain broad limits.

To simulate insect impact the stand must be grown from the time of planting. Initial basal area per acre is estimated within the program. Once a stand is established and an insect is chosen, the impact of that insect is determined. Control measures are available to counter the insect's impact.

Two stands are then grown simultaneously: one with insect control measures implemented (Stand 1), the other with no insect controls (Stand 2). Costs associated with insect monitoring and control and with stand establishment and management are discounted and compared with discounted revenues from thinnings and the final stand harvest.

Hardware Requirements

RPPEEST is designed to run on the IBM PC and compatible microcomputers. The program was developed on a Leading Edge Model D microcomputer. 64 kilobytes of RAM memory is sufficient to run the program. A monochrome monitor is required. Graphics capabilities are not required. Currently, no provision is made for printing RPPEST output.

Running RPPEST

RPPEST appears on the disk as a .COM file which has two overlay files associated with it. Overlay files are labeled RPPEST with a numeric file extension (eg., RPPEST.001). The overlay files must be present on the same disk drive as the RPPEST.COM file for the program to function correctly.

To start the program make sure the RPPEST disk is in the default disk drive, and type RPPEST followed by a carriage return.

Initializing RPPEST

The main menu of the program provides an opportunity to specify simulation of red pine growth and yield with or without the influence of insects. If insects are desired item 2 must be chosen at this point. Insects may not be introduced into the stand later in the program.

Initial Stand Information

Both choices from the main menu require that several pieces of information be entered initially. Some of this information is common to both choices, such as site index, merchantability limits, stand age, initial basal area, trees per acre, harvest age and display interval. These common items are described below.

Site index is the height of dominant and codominant trees, in feet, at the base age of 50 years. In RPPEST, site index must be between 30 and 90 feet. Minimum poletimber dbh is used to specify the smallest diameter of merchantable trees. This value must be larger than 3.5 inches in order to fall within the data range of the volume equations used. Minimum sawtimber dbh defines the smallest trees for which International 1/4 inch board foot volume will be calculated. The only restriction placed on this value is that it be larger than the minimum poletimber dbh.

Stand age at the beginning of projections may be any age less than the maximum age of 135 years if there are no insects in the stand. Insect infested stands must begin in the year of planting, year 0.

Initial basal area, in square feet per acre, must be set by the user if the trees are 25 years or older from seed. If the trees are younger than 25 and taller than 4.5 feet, initial basal area may be specified or estimated by the program using an equation developed by Lundgren (1981) from data originally collected by Wambach (1967). Initial basal area must be lower than 180 square feet per acre.

Trees per acre at the beginning of projections must be between 200 and 1600.

Harvest age must be no greater than the maximum age of 135 years.

The display interval is the number of years of stand projections between stand description displays.

Additional Initial Information

Additional initial information is needed for simulation of stands under insect attack. One of the four insects must be selected, and the user must enter information on type of control, desired site preparation, type of planting, product prices, and inventory and sale layout costs. Default costs are available in the program, or the user may enter a different value.

Site preparation may be selected as mechanical, chemical, or no site preparation. Mechanical and chemical methods have default costs of $64.57 and $52.30, respectively, adjusted for inflation (Winebar and Gunter, 1984). They are assumed to be equally effective.

Planting may be done by machine or by hand. Default adjusted costs from Winebar and Gunter (1984) for machine planting without site preparation are $134.41 per acre for less than 800 trees and $146.65 per acre for more than 800 trees. Hand planting without site preparation costs $124.25 per acre for less than 800 trees and $116.02 per acre for more than 800 trees.

Machine planting with site preparation costs $88.09 per acre for less than 800 trees, and $136.24 per acre for more than 800 trees. Hand planting with site preparation costs $109.50 per acre for less than 800 trees, and $148.97 per acre for more than 800 trees.

The default price for red pine cordwood is $12.00 per standard cord, while sawlogs are $70.00 per MBF (International 1/4" rule) (Norris, 1987).

The default cost for timber inventory is $6.73 per acre (Winebar and Gunter, 1984). An inventory is conducted before every thinning and before the final harvest.

The default cost for timber sale layout is $9.38 per acre (Winebar and Gunter, 1984). Sale layout occurs with every thinning and the final harvest.

Default forest management costs are summarized in Table 3.

The default annual interest rate used for discounting costs and revenues to the year of planting is set at 5% (R.J. Marty, 1987, Michigan State University, personal communication). This is a real rate, beyond inflation.

Table 3. Default costs for forest management activities and stumpage.

Management Activity	Default Cost
Mechanical Site Preparation	$ 64.57 /ac.
Chemical Site Preparation	$ 52.30 /ac.
Machine Planting without Site Prep, <800 trees / ac.	$134.41 /ac.
Machine Planting without Site Prep, >800 trees / ac.	$146.65 /ac.
Hand Planting without Site Prep, <800 trees / ac.	$124.25 /ac.
Hand Planting without Site Prep, >800 trees / ac.	$116.02 /ac.
Machine Planting with Site Prep, <800 trees / ac.	$ 88.09 /ac.
Machine Planting with Site Prep, >800 trees / ac.	$136.24 /ac.
Hand Planting with Site Prep, <800 trees / ac.	$109.50 /ac.
Hand Planting with Site Prep, >800 trees / ac.	$148.97 /ac.
Timber Inventory	$ 6.73 /ac.
Timber Sale Layout	$ 9.38 /ac.
Cordwood Price	$ 12.00 /cord
Sawtimber Price	$ 70.00 /MBF

Insect Management

RPPEST simulates the individual influence of four insects on the growth and yield of red pine: white grubs, pine root collar weevil, Saratoga spittlebug, red-headed pine sawfly. Ranges of random values used in RPPEST insect models are summarized in Table 4. Default insect management costs are summarized in Table 5.

White Grubs

White grub impact on red pine shows up as seedling mortality and site index reduction. Control is achieved by applying a pesticide at the time of planting. The default cost for control is $30.00 per acre; control occurs in year 0.

The plantation must be monitored for grub populations before control is attempted. The default cost for monitoring is $8.00 per acre, and it occurs in year 0.

If control is selected it is assumed to be 83% effective, and mortality is reduced by this factor. Site index is also returned to the original value. If no control is chosen mortality remains unaffected, there are no monitor or control costs recorded and site index remains reduced.

Table 4. Range of random values used in RPPEST simulations.

Insect	Action	Random Range
White Grubs	1st & 2nd year mortality	0% - 40%
	3rd year mortality	1% - 12%
	4th year mortality	0% - 3%
	Site index reduction	5 - 9 units
Pine root collar weevil	Hazard zone	low, medium, high
	Low hazard zone mortality	0% - 5%
	Medium hazard zone mortality	0% - 10%
	High hazard zone mortality	5% - 15%
Saratoga spittlebug moderate	Risk rating	low, high
	Low risk growth loss	0 - 4 years
	Moderate risk growth loss	4 - 10 years
	High risk growth loss	10 years to total stand
Red-headed pine sawfly	Outbreak year	6 - 13
	SRC I mortality	0% - 1.3%
	SRC II mortality	0% - 1.8%
	SRC III mortality	100%

Table 5. Default costs for insect management activities in RPPEST.

Insect	Activity	Default Cost
White grubs	Monitoring	$ 8.00 /ac.
	Chemical Control	$30.00 /ac.
Pine root collar weevil	Monitoring	$10.00 /ac.
	Pruning & Scraping	$ 6.00 /ac.
	Basal Spraying	variable
Saratoga spittlebug	Risk Rating	$ 1.00 /ac.
	Monitoring	$ 1.50 /ac.
	Aerial Insecticide	$12.00 /ac.
	Herbicide-Hand Planting	$70.00 /ac.
	Herbicide-Machine Planting	$60.00 /ac.
Red-headed pine sawfly	Monitoring	$ 1.50 /ac.
	Aerial Insecticide	$12.00 /ac.
	Herbicide-Hand Planting	$70.00 /ac.
	Herbicide-Machine Planting	$60.00 /ac.

Pine Root Collar Weevil

Pine root collar weevil causes mortality in red pine plantations which have not reached crown closure. The default cost of monitoring weevil populations is $10.00 per acre and occurs in years 3 and 6 if control is chosen.

Weevils may be controlled by basal pruning and by scraping away duff and topsoil from each tree. The default cost for pruning and scraping assumes a labor rate of $6.00 per hour and a work rate of 60 trees per hour. This action occurs in year 7. Weevils may also be controlled by basal insecticide spraying of each tree. The default cost assumes a labor rate of $6.00 per hour, a work rate of 120 trees per hour and a chemical cost of $5.00 per acre. This cost occurs in year 7.

Saratoga Spittlebug

Saratoga spittlebug impact on red pine manifests itself in years of lost growth and tree mortality in stands which have not reached crown closure.

One method of controlling spittlebug is to reduce alternate host plants by applying herbicide at planting. The default cost for this control method is $70.00 per acre if the stand is planted by hand, and $60.00 per acre if the stand is machine planted. This cost occurs in year 0.

Risk rating the plantation costs $1.00 per acre and also occurs in year 0.

A second control method is to reduce the spittlebug population using insecticide. The default cost is $12.00 per acre for aerial spraying (Heyd, 1982). Monitoring populations costs $1.50 per acre. The number of sprays and monitorings depends upon the hazard rating of the stand and site index (Heyd, 1985). Low risk and site index greater than 50 requires monitoring in years 4 and 6, and spraying in year 6. If site index is less than 50, an additional monitoring is scheduled for year 10. Moderate risk rating and site index greater than 50 requires monitoring in years 4, 6, and 10, and spraying in year 6. Moderate risk and site index less than 50 requires monitoring in years 4, 6, 10, and 12, and sprays in years 6 and 12. High risk and site index higher than 50 requires monitoring in years 4, 8, and 9, and sprays in years 4 and 9. High risk and site index less than 50 requires monitoring in years 4, 8, 9, and 14, and sprays in years 4, 9, and 14. All controls are assumed to put the stand in the low risk category.

Red-Headed Pine Sawfly

Red-headed pine sawfly causes tree mortality and deformation in red pine stands on poor sites (site index < 50) which have not reached crown closure.

Sawfly impact can be greatly reduced by controlling competing vegetation or by spraying insecticide to reduce sawfly populations. Controlling competing vegetation occurs in year 0 and has a default cost of $70.00 per acre if the stand was planted by hand, or a cost of $60.00 per acre if machine planted. Controlling sawfly populations occurs during the randomly determined outbreak year and costs $12.00 per acre. Monitoring sawfly populations occurs 3 years before the outbreak year and in the outbreak year, and costs $2.00 per acre. Both control methods are assumed to reduce impact to a negligible level.

Thinning

Thinning in RPPEST assumes every tree has an equal chance of being selected, as in mechanical thinning. Trees are removed equally across the entire diameter distribution. Basal area per acre and trees per acre are reduced by the same fraction, leaving quadratic mean diameter unchanged.

To implement thinning requires initial values for age at first thinning, minimum basal area at the first thinning, thinning interval, stand age at final thinning. Also, one of three types of thinnings must be chosen: remove a specific number of trees per acre, remove a constant proportion of the basal area or trees per acre, or thin to a specified residual basal area per acre.

Economic Evaluation

To compare costs and revenues which occur throughout a lengthy rotation on an even basis these values must be discounted to a common year. In this program that is the year of planting, year 0. Gunter and Haney (1984) suggest that for ease of calculation all costs and revenues should be considered to occur at the end of the year. This convention is followed in RPPEST.

Two analyses are performed after Drapek (1985). The first compares all costs with all revenues to evaluate the overall management strategy using Net Present Value (NPV). Net Present Value is calculated by subtracting the sum of the discounted costs from the sum of the discounted revenues. If the result of this subraction is positive then the red pine stand in question compares favorably with alternative investments.

The second analysis also uses NPV to evaluate insect management efforts. The discounted value of the wood saved by monitor and control techniques is compared with the discounted costs of those techniques. Again, a positive NPV indicates a worthwhile investment.

A Sample RPPEST Program Run

Figures 2 through 16 show a typical RPPEST run involving white grubs. Figures 2 through 6 show preliminary information. The remainder of the figures show periodic output and final summary screens.

```
1.  Site index = 60

2.  Minimum poletimber dbh = 5.0

3.  Minimum sawtimber dbh = 10.0

4.  Seedling age at planting = 2

5.  Stand age at beginning of projections = 0

6.  Initial basal area = 0.0

7.  Trees per acre = 680

8.  Harvest age = 50

9.  Display interval (years) = 10

Do you want to change any of these values   (Y or N)?   N
```

Figure 2. Initial Stand Information.

```
1.  The insect of interest is White Grubs

2.  Site preparation is going to be Mechanical

3.  Site preparation cost per acre is $  64.57

4.  The trees will be planted by  Machine

5.  Planting cost per acre is $  88.09

6.  Annual interest rate is  5.0 percent.

7.  Sale layout cost per acre is $  9.38

8.  Inventory cost per acre is $  6.73

9.  Price / cord is $  12.00  and price / MBF is $  70.00


Do you want to change any of these values   (Y or N)?   N
```

Figure 3. Initial Site Preparation and Economic Information.

The winds of fate have decreed that your 1st year survival rate will be 78.8 %.

Your plantation now contains 536 trees / acre.

Press the Space Bar to continue.

Figure 4. First Year Seedling Survival.

You have chosen to simulate the effects of white grubs on projected red pine volume yields. A randomly generated population of 0.35 grubs per cubic foot of soil has been established. This number may vary between 0 and 0.75.

A combined 1st and 2nd year seedling mortality of 10 percent has been set to correspond to this population level.

Seedling mortality in year 3 will be 2 percent.
Seedling mortality in year 4 will be 1 percent.

There are two insect management alternatives of interest in this case.

1. Apply a pesticide at the time of planting.
2. Plant the site and ignore the grubs, and incur no immediate cost.

Enter the number of your grub control choice.1

Figure 5. Insect Caused Seedling Mortality.

You may enter a cost of grub control you have in mind or accept the default value of $30.00 per acre.

Enter the cost (xx.xx) or enter 0 to use the default. 0

The proposed planting site must be monitored before control actions can begin. The default cost of monitoring is $8.00 per acre.

Enter the cost or enter 0 to use the default value. 0

Figure 6. Insect Monitor and Control Costs.

Stand 1 trees/ac = 524   Stand 1 harvest age = 50   Stand 1 SI = 60
Stand 2 trees/ac = 464   Stand 2 harvest age = 50   Stand 2 SI = 51
Minimum pole dbh = 5.0   Minimum sawtimber dbh = 10.0

Stand 1: Per Acre Description at the Beginning of Year 20

White Grubs				Poletimber		Sawtimber		Pole + Saw			
Dom Ht.	Trees /ac.	BA	Ave. dbh	Cu. ft.	Cords	Cu. ft.	MBF	Cu. ft.	MAI	Cords	MAI
25	524	133	6.7	979*	12.4*	43*	0.1**	1022*	53.8	12.9*	0.68
				1278^		56^		1335^	70.3^		

Total Non-Insect-Killed Trees = 0

Stand 2: Per Acre Description at the Beginning of Year 20

Dom Ht.	Trees /ac.	BA	Ave. dbh	Cu. ft.	Cords	Cu. ft.	MBF	Cu. ft.	MAI	Cords	MAI
21	464	105	6.3	659*	8.3*	11*	0.0**	670*	35.3	8.5*	0.45
				861^		14^		875^	46.1^		

Total Non-Insect-Killed Trees = 0

*3 & **6 in. min. top diams. inside bark.   ^ Tot stem vol, no top diam lim.

Press the Space Bar to continue.

Figure 7. Stand Description Display at Age 20.

Stand 1 trees/ac = 524 Stand 1 harvest age = 50 Stand 1 SI = 60
Stand 2 trees/ac = 464 Stand 2 harvest age = 50 Stand 2 SI = 51
Minimum pole dbh = 5.0 Minimum sawtimber dbh = 10.0

White Grubs				Poletimber		Sawtimber		Pole + Saw			
Dom Ht.	Trees /ac.	BA	Ave. dbh	Cu. ft.	Cords	Cu. ft.	MBF	Cu. ft.	MAI	Cords	MAI
Stand 1: Per Acre Description at the Beginning of Year 30											
38	524	187	8.0	1711* 2235^	21.7*	503* 657^	1.8**	2214* 2892^	76.3 99.7^	28.0*	0.97
Total Non-Insect-Killed Trees = 0											
Stand 2: Per Acre Description at the Beginning of Year 30											
32	464	154	7.7	1276* 1667^	16.1*	252* 330^	0.9**	1528* 1996^	52.7 68.8^	19.3*	0.67
Total Non-Insect-Killed Trees = 0											

*3 & **6 in. min. top diams. inside bark. ^ Tot stem vol, no top diam lim.

Press the Space Bar to continue.

Figure 8. Stand Description Display at Age 30.

Stand 1 trees/ac = 524 | Stand 1 harvest age = 50 | Stand 1 SI = 60
Stand 2 trees/ac = 464 | Stand 2 harvest age = 50 | Stand 2 SI = 51
Minimum pole dbh = 5.0 | Minimum sawtimber dbh = 10.0

Stand 1 Per Acre Description at the Beginning of Year 35

White Grubs				Poletimber		Sawtimber		Pole + Saw			
Dom Ht.	Trees /ac.	BA	Ave. dbh	Cu. ft.	Cords	Cu. ft.	MBF	Cu. ft.	MAI	Cords	MAI
44	524	211	8.5	1861*	23.6*	1047*	4.3**	2908*	85.5	36.8*	1.08
CUT	-276	-111	8.5	-970	-12.3	-562	-2.33	-1532		-19.4	
44	248	100	8.5	891	11.3	485	2.01	1376		17.4	

Stand 2 Per Acre Description at the Beginning of Year 35

Dom Ht.	Trees /ac.	BA	Ave. dbh	Cu. ft.	Cords	Cu. ft.	MBF	Cu. ft.	MAI	Cords	MAI
37	464	179	8.3	1432*	18.1*	633*	2.6**	2065*	60.7	26.1*	0.77
CUT	-204	-79	8.3	-636	-8.1	-271	-1.12	-907		-11.5	
37	260	100	8.3	796	10.1	362	1.50	1157		14.7	

Press the Space Bar to continue.

Figure 9. Stand Description After Thinning at Age 35.

```
1.  Thinning Method:  Constant residual basal area per acre.

2.  Residual BA per acre:  100.0

3.  Age at first thinning:  35

4.  Minimum BA / acre before thinning begins:  0.0

5.  Thinning interval:  10

6.  Age at last thinning:  45


Do you want to change any of these values  (Y or N)?
```

Figure 10. Thinning Parameter Change Menu.

Stand 1 trees/ac = 524 — Stand 1 harvest age = 50 — Stand 1 SI = 60
Stand 2 trees/ac = 464 — Stand 2 harvest age = 50 — Stand 2 SI = 51
Minimum pole dbh = 5.0 — Minimum sawtimber dbh = 10.0

Stand 1: Per Acre Description at the Beginning of Year 50

White Grubs				Poletimber		Sawtimber		Pole + Saw			
Dom Ht.	Trees /ac.	BA	Ave. dbh	Cu. ft.	Cords	Cu. ft.	MBF	Cu. ft.	MAI	Cords	MAI
60	162	124	11.6	86*	1.1*	2167*	13.9**	2253*	46.0	28.5*	0.58
				113^		2831^		2944^	60.1^		

Total Non-Insect-Killed Trees = 0

Stand 2: Per Acre Description at the Beginning of Year 50

51	177	121	11.0	169*	2.1*	1710*	10.9**	1879*	38.3	23.8*	0.49
				221^		2234^		2454^	50.1^		

Total Non-Insect-Killed Trees = 0

*3 & **6 in. min. top diams. inside bark. ^ Tot stem vol, no top diam lim.

Press the Space Bar to continue.

Figure 13. Stand Description Display at Age 50.

```
                Stand 1 Volume Summary (with insect management)
             ---------------------------------------------------

Harvested merchantable cordwood volume =                     28.5 cords/acre.
Merchantable cordwood volume removed in thinnings =          30.6 cords/acre.
Total merchantable cordwood volume produced =                59.1 cords/acre.
MAI for merchantable cordwood volume =                       1.18 cords/acre/year.

Harvested International 1/4" bd. ft. volume =                13.9 MBF/acre.
Int. 1/4" bd. ft. volume removed in all thinnings =          6.5 MBF/acre.
Total International 1/4" bd. ft. volume produced =           20.4 MBF/acre.
MAI for International 1/4" bd. ft. volume =                  0.41 MBF/acre/year.

Cordwood poles harvested in addition to sawtimber =          1.1 cords/acre.
Cordwood poles removed in all thinnings =                    14.2 cords/acre.
Total pole volume produced in addition to sawtimber =        15.3 cords/acre.

Harvested merchantable cubic foot volume =                    2253 cu ft/acre.
Merchantable cubic foot volume removed in thinnings =         2418 cu ft/acre.
Total merchantable cubic foot volume produced =               4672 cu ft/acre.
MAI for merchantable cubic foot volume =                     93.4 cu ft/acre/year.


                   Press the Space Bar to continue.
```

Figure 14. Stand 1 Volume Summary.

```
                    Final Economic Analysis
                   ---------------------------

Insect Managed Stand
--------------------

Discounted value of total cordwood volume =                $ 85.59
Discounted value of sawtimber plus pole cordwood volume =  $ 173.08
Discounted value of all costs =                            $ 195.37

Net present value of the stand in year of planting (year 0) is:
     Cordwood NPV = $ -109.78      Sawtimber NPV = $ -22.29

Evaluation of Insect Management Efforts
---------------------------------------
Discounted value of cordwood volume saved =                $ 25.58
Discounted value of sawtimber volume saved =               $ 50.54
Discounted value of insect management costs =              $ 38.00

Discounted value of insect management efforts :
     Cordwood NPV = $ -12.42       Sawtimber NPV = $ 12.54

                   Press the Space Bar to continue.
```

Figure 16. Final Economic Analysis.

List of References

Drapek, R.J. 1985. A simulation of jack pine budworm and white pine weevil monitor and control techniques and their effects on jack pine economics. M.S. Thesis, 118 p. Michigan State University, E. Lansing.

Gunter, J.E. and H.L. Haney. 1984. Essentials of Forestry Investment Analysis. 337 p. OSU Bookstores, Corvallis, OR.

Heyd, R.L. 1982. Insects versus artificial regeneration of conifers. P. 358-374 in G.D. Mroz and J.F. Berner(compilers). Proc. Artif. Regen. of Conifers in the Upper Great Lakes Region, 26-28 Oct. 1982, Green Bay, WI, 435 p. Mich. Tech. Univ., Houghton.

Heyd, R.L. 1985. NPVUNPL. A Saratoga spittlebug simulation program. Available from the Michigan Cooperative Forest Pest Management Program. East Lansing, Michigan.

Lundgren, A.L. 1981. The effect of initial number of trees per acre and thinning densities on timber yields from red pine plantations in the Lake States. USDA Forest Service Res. Paper NC-193, 25 p.

Norris, F.W. 1987. Timber Mart North, 1st quarter 1987. Highlands, NC.

Wambach, R.F. 1967. A silvicultural and economic appraisal of initial spacing in red pine. Ph.D. Dissertation, 282 p. University of Minnesota, Minneapolis.

APPENDIX B

APPENDIX B

A Program Listing of RPPEST

```
  TYPE
    regpack = RECORD
            CASE Boolean OF
              True : (AX, BX, CX, DX, BP, SI, DS, DI, ES, Flags :
                      Integer);
              False : (AL, AH, BL, CL, CH, DL, DH : Byte);
              End;
    charset = set of char;
    p2 = array [1..40] of integer;
    string18 = string[22];


CONST
    AutoUpCase: boolean = false;   {F. ConIn }
    range: charset = [#0..#255];   {Allowable input range F. ConIn }
    special: charset = [#13,#8];   {F. ConIn }

     zp = 0.2316419;              MechPrep = 64.57;
    zb1 = 0.31938153;             ChemPrep = 52.30;
    zb2 = -0.356563782;
    zb3 = 1.781477937;
    zb4 = -1.821255978;
    zb5 = 1.330274429;

    StandArea = 1;


VAR
     A: p2;
    AAveBA: real;
    ADomHt: integer;
    AgeAtEst: integer;
    AllCostsDiscounted: real;
    AMBFVolInt: real;
    AMBFVolScrib: real;
    AMerchCFV: real;
    AMerchCords: real;
    AMerchPoleCFV: real;
    AMerchSawCFV: real;
    AMortalityBA: real;
    AMortalityClass: p2;
    AMortalityTPA: integer;
    ANMCFV: real;
    ANMSawBF: real;
    ANMSawCFV: real;
    APoleCFV: real;
    APoleClass: integer;
    APoleCords: real;
    ASawCFV: real;
    ASawClass: integer;
    ASawCords: real;
    ASmallCFV: real;
    ATotalCFV: real;
    ATPA: integer;
    AveBA:real;
    AveClass: integer;
    BA: real;
    BAC: array [1..40] of real;
    BAGrowth1: real;
    BAGrowth2: real;
    BAProToRemove: real;
    BAToLeave: real;
```

```
BAUnder25: real;
BB: array [1..40] of real;
BFperCF: real;
BHA: real;
BHA1: real;
BiggestClass: integer;
BTPA: integer;
BugAveBA: real;
BugBA: real;
BugBAC: array [1..40] of real;
BugBAGrowth1: real;
BugBAGrowth2: real;
BugBAUnder25: real;
BugBFperCF: real;
BugBHA: real;
BugBHA1: real;
BugBiggestClass: integer;
BugChoice: integer;
BugClass: p2;
BugDGR: real;
BugDiamSD: real;
BugDiscountedCordsRevenue: real;
BugDiscountedSawRevenue: real;
BugDomHt: real;
BugHarvestAge: integer;
BugMaxBAIncr: real;
BugMaxDiamIncr: real;
BugMBFVolInt: real;
BugMBFVolScrib: real;
BugMerchCFV: real;
BugMerchCords:real;
BugMerchPoleCFV: real;
BugMerchSawCFV: real;
BugMortalityBA: real;
BugMortalityClass: p2;
BugMortalityTPA: integer;
BugMortalityTreeCount: integer;
BugMortVolSum: real;
BugNMcfv: real;
BugNMSawBF: real;
BugNMSawcfv: real;
BugPoleBA: real;
BugPoleCFV: real;
BugPoleCords: real;
BugQMDbh: real;
BugSawBA: real;
BugSawCFV: real;
BugSawCords: real;
BugSI: integer;
BugSmallBA: real;
BugSmallCFV: real;
BugSmallestClass: integer;
BugThinnedBA: real;
BugThinnedBAC: array [1..40] of real;
BugThinnedClass: p2;
BugThinnedPoleBA: real;
BugThinnedSawBA: real;
BugThinnedSmallBA: real;
BugThinnedTPA: integer;
BugTMBFVolInt: real;
BugTMBFVolScrib: real;
BugTMerchcfv: real;
BugTMerchCords: real;
BugTMerchPolecfv: real;
BugTMerchSawcfv: real;
BugTotalCFV: real;
BugTPA: integer;
BugTPolecfv:real;
BugTPoleCords: real;
BugTSawcfv: real;
BugTSawCords: real;
BugTSmallcfv: real;
```

```
BugTThinnedTPA: integer;
BugTTMBFVolInt: real;
BugTTMerchCords: real;
BugTTotalcfv: real;
BugTTPoleCords: real;
BugTTTotalCFV: real;
CBAUnder25: real;
Choice1: integer;
Class: p2;
CMortalityBA: real;
CMortalityTPA: integer;
CPoleBA: real;
CSawBA: real;
CSmallBA: real;
CTPA: integer;
DGR: real;
DiamSD: real;
DiscountedBugCosts: real;
DiscountedCordsRevenue: real;
DiscountedSawRevenue: real;
DiscountedTotalCosts: real;
DiscountedValue: real;
DisplayInterval: integer;
DomHt: real;
GAClass: p2;
GCCost: real;
GMCost: real;
GrubControlChoice: integer;
GrubControlCost: real;
GrubMonitorCost: real;
GrubMortPercent: real;
GrubNumbers: real;
HarvestAge: integer;
HighGrubMort: real;
HPCost: real;
InsectControlCostsDiscounted: real;
InterestRate: real;
InventoryCost: real;
LayOutCost: real;
Lowclass: integer;
LowGrubMort: real;
MaxBAIncr: real;
MaxDiamIncr: real;
MaxTAge: integer;
MBFVolInt: real;
MBFVolScrib: real;
MerchCFV: real;
MerchCords:real;
MerchPoleCFV: real;
MerchSawCFV: real;
MinTAge: integer;
MinTBA: real;
Mort1: real;
Mort2: real;
MortalityBA: real;
MortalityClass: p2;
MortalityTPA: integer;
MortalityTreeCount: integer;
MortalityTrees: integer;
MortVolSum: real;
MPCost: real;
N: real;
NextScheduledThin: integer;
NMcfv: real;
NMSawBF: real;
NMSawcfv: real;
OutbreakYear: integer;
PlantingCost: real;
PlantingMethod: integer;
PoleBA: real;
PoleCFV: real;
PoleClass: integer;
```

```
PoleClassUp: real;
PoleCords: real;
PoleMin: real;
PricePerCord: real;
PricePerMBF: real;
QMDbh: real;
ReducedTPA: integer;
regs: regpack;
ROChoice: integer;
SawBA: real;
SawCFV: real;
SawClass: integer;
SawClassUp: real;
SawCords: real;
SawflyControlCost: real;
SawflyMonitorCost: real;
SawMin: real;
SBugChoice: String18;
SeedAge: integer;
SFCCost: real;
SHazardZone: string[6];
SI: integer;
SID: string[10];
SIReducer: integer;
SitePrep: integer;
SitePrepCost: real;
SmallBA: real;
SmallCFV: real;
SmallestClass: integer;
SpittlebugControlChoice: integer;
SPlantingMethod: String18;
SRC1Percent: real;
SRC2Percent: real;
SRC3Percent: real;
SRiskRating: string[8];
SS: real;
SSitePrep: String18;
SSm: real;
StandAge: integer;
StartAge: integer;
StartBA: real;
StartTPA: integer;
StartTPA1: integer;
StartTPA2: integer;
StorePtr: integer;
SurvivalModifier: real;
SurvivalRate: real;
ThinChoice: char;
ThinnedBA: real;
ThinnedBAC: array [1..40] of real;
ThinnedClass: p2;
ThinnedPoleBA: real;
ThinnedSawBA: real;
ThinnedSmallBA: real;
ThinnedTPA: integer;
ThinningInterval: integer;
ThinningType: integer;
Time: integer;
TMBFVolInt: real;
TMBFVolScrib: real;
TMerchcfv: real;
TMerchCords: real;
TMerchPolecfv: real;
TMerchSawcfv: real;
TotalCFV: real;
TPA: integer;
TPolecfv:real;
TPoleCords: real;
TreesToRemove: integer;
TSawcfv: real;
TSawCords: real;
TSmallcfv: real;
```

```
TThinnedTPA: integer;
TTMBFVolInt: real;
TTMerchCords: real;
TTotalcfv: real;
TTPoleCords: real;
TTTotalCFV: real;
UpClass: integer;
WeevilControlCost: real;
WeevilMonitorCost: real;
WMort: real;
X6: real;
X7: real;
X8: real;
X9: real;
Year3GrubMort: real;
Year4GrubMort: real;

     r: array [0..135] of
                 record
                   AveBA: real;
                   BA: real;
                   DGR: real;
                   DomHt: real;
                   MBFVolInt: real;
                   MBFVolScrib: real;
                   MerchCFV: real;
                   MerchCords: real;
                   MerchPoleCFV: real;
                   MerchSawCFV: real;
                   MortalityBA: real;
                   MortalityTPA: integer;
                   NMCFV: real;
                   PoleCFV: real;
                   PoleCords: real;
                   QMdbh: real;
                   SawCFV:real;
                   SawCords: real;
                   SmallCFV: real;
                   StandAge: integer;
                   TotalCFV: real;
                   TPA: integer;
                  end;


bugr: array [0..135] of
                 record
                   AveBA: real;
                   BA: real;
                   DGR: real;
                   DomHt: real;
                   MBFVolInt: real;
                   MBFVolScrib: real;
                   MerchCFV: real;
                   MerchCords: real;
                   MerchPoleCFV: real;
                   MerchSawCFV: real;
                   MortalityBA: real;
                   MortalityTPA: integer;
                   NMCFV: real;
                   PoleCFV: real;
                   PoleCords: real;
                   QMdbh: real;
                   SawCFV:real;
                   SawCords: real;
                   SmallCFV: real;
                   StandAge: integer;
                   TotalCFV: real;
                   TPA: integer;
                 end;

       t: array [0..135] of
```

```
                record
                  StandAge: integer;
                  ThinnedBA: real;
                  ThinnedTPA: integer;
                  TMBFVolInt: real;
                  TMBFVolScrib: real;
                  TMerchCFV: real;
                  TMerchCords: real;
                  TMerchPoleCFV: real;
                  TMerchSawCFV: real;
                  TPoleCFV: real;
                  TPoleCords: real;
                  TSawCFV:real;
                  TSawCords: real;
                  TSmallCFV: real;
                  TTotalCFV: real;
                end;


        bugt: array [0..135] of
                record
                  StandAge:integer;
                  ThinnedBA: real;
                  ThinnedTPA: integer;
                  TMBFVolInt: real;
                  TMBFVolScrib: real;
                  TMerchCFV: real;
                  TMerchCords: real;
                  TMerchPoleCFV: real;
                  TMerchSawCFV: real;
                  TPoleCFV: real;
                  TPoleCords: real;
                  TSawCFV:real;
                  TSawCords: real;
                  TSmallCFV: real;
                  TTotalCFV: real;
                end;


       econ: array [0..135] of
                 record
                   Control: real;
                   Inventory: real;
                   Layout: real;
                   Monitor: real;
                 end;


  Procedure Randomize(I,J: Integer);

var
  RSet    : record
              AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags: Integer;
            end;

  Ch      : Char;

begin
  if (I=0) and (J=0) then begin     { Generate a random random number
seed }
    RSet.AX:=$2C00;                                 { DOS time of day
function }
    MSDos(RSet);
    I:=RSet.CX;                                { Set I and J to the system
time }
    J:=RSet.DX;
    Delay(100);   { This delay may have to be increased for faster
systems }
    MSDos(RSet);
    if (I=RSet.CX) and (J=RSet.DX) then begin        { Clock isn't
ticking }
      I := 0;
```

```
      J := 0;
      while KeyPressed do
        Read(Kbd,Ch);                               { Clear keyboard
buffer }
      Write('Hit any key to set the random number generator: ');
      repeat
        I := I+13;
        J := J+17
      until Keypressed;
      Read(Kbd,Ch);                                  { Absorb the
character }
      WriteLn
    end;
  end;
  MemW[DSeg:$129]:=I;  { This is the core of the routine: store a 32
bit }
  MemW[DSeg:$12B]:=J;  {  seed at locations DSeg:$0129...DSeg:$012C
 }
end; { of procedure Randomize }


 Function ConIn : char;
    var myCH: char;
    BEGIN
      WITH regs DO
        BEGIN
          repeat
            AH :=8;
            MSDOS(regs);
            myCH :=chr(AL);
          until myCH in (range + special);
        END;
      IF AutoUpCase THEN myCH := UpCase(myCH);
      conin := myCH;
    END;


   Procedure SetSI;            {called from procedure InitialStandInfo}

     begin
       repeat
       clrscr; gotoxy(1,10);delline; gotoxy(5,10);
       write('Enter site index for this stand   (base age 50 yrs.).
');
       range:= ['0'..'9'];
       readln(SI);
       until SI in [30..90];
       BugSI := SI;
     end;        {of SetSI}


   Procedure SetPoleMin;     {called from procedure InitialStandInfo}

     begin
       repeat
       clrscr;gotoxy(1,10); delline; gotoxy(5,10);
       write('Choose a minimum poletimber dbh, greater than 3.5 inches.
 ');
       range:= ['0'..'9','.'];
       readln(PoleMin);
       until PoleMin>3.5;
     end;     {of SetPoleMin}


   Procedure SetSawMin;         {called from procedure
InitialStandInfo}

     begin
       clrscr;
       repeat
       gotoxy(1,10); delline; gotoxy(5,10);
       write('Choose a minimum sawtimber dbh  (inches).   ');
```

```
      range:= ['0'..'9','.'];
      readln(SawMin);
      until (SawMin > PoleMin);
    end;      {of SetSawMin}


  Procedure SetEstablishAge;        {called from InitialStandInfo}

    begin
      clrscr;
      repeat
      gotoxy(1,10); delline; gotoxy(5,10);
      write('How old were the seedlings at the time of planting
(years)?   ');
      range:= ['0'..'9'];
      readln(AgeAtEst);
      until AgeAtEst >=0;
    end;       {of SetEstablishAge}


  Procedure SetAgeAtStart;            {called from InitialStandInfo}

    begin
    case Choice1 of
    1: begin
      clrscr; gotoxy(5,10);
      writeln('Stand age begins at the time of stand establishment
      (year 0).');
      repeat
      gotoxy(1,13); delline; gotoxy(5,13);
      write('At what stand age do you want to start projections?   ');
      range:= ['0'..'9'];
      readln(StartAge);
      until StartAge >=0;
      end;
   2: StartAge := 0;
   end;   {of case statement}
      SeedAge:= AgeAtEst + StartAge; StandAge:= StartAge;
   end;      {of SetAgeAtStart}


  Procedure SetHarvestAge;              {called from InitialStandInfo}


    begin
      clrscr;repeat
      gotoxy(1,10); delline; gotoxy(5,10);
      write('At what stand age do you want to harvest this stand?
      ');
      range:=['0'..'9'];
      readln(HarvestAge);
      until (HarvestAge>StartAge) and (HarvestAge <= 125);
      BugHarvestAge := HarvestAge;
      end;           {of SetHarvestAge}


  Procedure SetDisplayInterval;         {called from InitialStandInfo}


    begin
      clrscr;gotoxy(1,10); delline; gotoxy(5,10);
      write('At what interval (years) do you want to display stand
  statistics?   ');
      range:=['0'..'9'];
      readln(DisplayInterval);
      end;          {of SetDisplayInterval}

  Procedure SetTPA;             {called from InitialStandInfo}

    begin
      case StartAge of
        0..10: begin
```

```
            clrscr;repeat
            clrscr; gotoxy(5,10);
            if StartAge = 0 then write('Enter the number of
            seedlings per acre in the stand.   ')
            else
            write('Enter the number of established trees per acre
            in the stand.   ');
            range:= ['0'..'9'];
            readln(StartTPA);
            if StartTPA < 200 then
              begin
              gotoxy(5,15);writeln('Initial number of trees per
              acre must be greater than 200.');
              delay(3000);
              end;
            until StartTPA >= 200;
            TPA := StartTPA;
            BugTPA := StartTPA;
            end;
            else
            begin
            clrscr;
            repeat
            gotoxy(1,10); delline; gotoxy(5,10);
            write('Enter the number of established trees per acre
            in the plantation.   ');
            range:= ['0'..'9'];
            readln(StartTPA);
            until StartTPA < 1600;
            TPA := StartTPA;
            BugTPA := StartTPA;
            end;
      end;  {of case}
  end;   {of SetTPA}


Procedure BAU(ASI:integer; ATPA:integer);
                                {Lundgren, 1981;  stand age < 25
                                                           yrs.}
  begin
    CBAUnder25 := 6.565 * ASI * (exp(1.1677 * ln(1-(exp(-0.04018 *
    SeedAge))))) * (1-exp(-0.001885 * ATPA));
  end;

Procedure BAU1;
  begin
    BAU(SI,TPA);
    BAUnder25 := CBAUnder25;
  end;

Procedure BugBAU;
  begin
    BAU(BugSI,BugTPA);
    BugBAUnder25 := CBAUnder25;
  end;


Procedure BHAge;
  begin
    BHA1 := SeedAge - (10.5 + (0.05 * SI));
    if BHA1 <= -1 then BHA := 0
    else BHA := BHA1;
  end;

Procedure BugBHAge;
  begin
    BugBHA1 := SeedAge - (10.5 + (0.05 * BugSI));
    if BugBHA1 <= -1 then BugBHA := 0
    else BugBHA := BugBHA1;
  end;
```

```
Procedure SetBA;              {called from InitialStandInfo}
  var ch:char;
  begin
  clrscr;
  BHAge;
  case Seedage of
     1..24: begin
          if  BHA1 <= -1 then begin
            StartBA :=0;
            BA := StartBA;
            BugBA := StartBA;
          end
          else begin
             gotoxy(5,10); writeln('You may specify a basal area per
             acre for this stand.  If you choose ');
             gotoxy(5,11);
             writeln('not to, the program will estimate it for
             you.');
             gotoxy(5,14);
             write('Do you want to specify the basal area per acre?
             (Y or N)    ');
             range:= ['y','n','Y','N'];
             readln(ch);
             if (ch = 'Y') or (ch = 'y') then
               begin
               repeat
               gotoxy(1,14); delline; gotoxy(5,14);
               write('Enter basal area of the stand in square feet
               per acre.    ');
               range:= ['0'..'9','.'];
               readln(StartBA);
               until (StartBA >0) and (StartBA < 180);
               BA := StartBA;
               BugBA := StartBA;
               end
              else begin
               BAU1; BugBAU;
               StartBA:= BAUnder25;
               BA := StartBA;
               BugBA := StartBA;
               end;
        end;    { of else from BHA1 < -1  if - then}
     end;    {of case 1..24}
    25..250: begin
              repeat
              gotoxy(1,10); delline; gotoxy(5,10);
              write('Enter the basal area of the stand in square feet
              per acre.    ');
              range:= ['0'..'9','.'];
              readln(StartBA);
              until (StartBA >= 30);
              BA := StartBA;
              BugBA := StartBA;
              end;
    end;    {of case}
  end;   {of SetBA}


Procedure DisplayISI;                  {called from EditISI}

  begin
  clrscr;
  gotoxy(5,5); writeln('1.  Site index = ',SI);
  gotoxy(5,7); writeln('2.  Minimum poletimber dbh = ',PoleMin:2:1);
  gotoxy(5,9); writeln('3.  Minimum sawtimber dbh = ',SawMin:2:1);
  gotoxy(5,11); writeln('4.  Seedling age at planting = ',AgeAtEst);
  gotoxy(5,13); writeln('5.  Stand age at beginning of projections =
   ',StartAge);
  gotoxy(5,15); writeln('6.  Initial basal area = ',BA:3:1);
  gotoxy(5,17); writeln('7.  Trees per acre = ',TPA);
  gotoxy(5,19); writeln('8.  Harvest age = ',HarvestAge);
  gotoxy(4,21); writeln('9.  Display interval (years) =
```

```
',DisplayInterval);
    end;        {of DisplayISI}


  Procedure EditISI;                  {called from Basics}
    var      Choice:char;    Which:integer;

    begin
    repeat
    DisplayISI; gotoxy(5,24);
    write('Do you want to change any of these values   (Y or N)?   ');
    range:= ['n','y','N','Y'];
    readln(Choice);
    if upcase(Choice) = #89 then
      begin
      repeat
      gotoxy(1,24); delline; gotoxy(5,24);
      write('Enter the number of the part to change.   ');
      range:= ['1'..'9'];
      readln(Which);
      until (Which>=1) and (Which<=10);
    case Which of
      1: SetSI;
      2: SetPoleMin;
      3: SetSawMin;
      4: SetEstablishAge;
      5: SetAgeAtStart;
      6: SetBA;
      7: SetTPA;
      8: SetHarvestAge;
      9: SetDisplayInterval;
    end;      {of case}
    end;
  until  (Choice = #78) or (Choice = #110);
  end;    {of EditISI}


  Procedure ChooseBug;
     var     choice: char;

    begin
      clrscr;gotoxy(5,5);
      write('1.  White grubs.');  gotoxy(5,8);
      write('2.  Pine root collar weevil.');  gotoxy(5,11);
      write('3.  Saratoga spittlebug.');  gotoxy(5,14);
      write('4.  Red-headed pine sawfly.');
      repeat
      gotoxy(1,19); delline; gotoxy(10,19);
      write('Enter the number of the insect that bugs you most.  ');
      range := ['1'..'4'];
      readln(BugChoice);
      until BugChoice in [1..4];
      case BugChoice of
        1: SBugChoice := 'White Grubs';
        2: SBugChoice := 'Root Collar Weevil';
        3: SBugChoice := 'Saratoga Spittlebug';
        4: SBugChoice := 'Red-headed Sawfly';
      end;   {of case}
    end;   {of ChooseBug}


  Procedure SetSitePrep;
    begin
      clrscr; gotoxy(5,5);
      write('1.  Mechanical site preparation.'); gotoxy(5,7);
      write('2.  Chemical site preparation.'); gotoxy(5,9);
      write('3.  No site preparation.');
      repeat
      gotoxy(1,15); delline; gotoxy(10,15);
      write('Enter the number of your choice of site preparation.
        ');
      range := ['1'..'3'];
```

```
    readln(SitePrep);
    until SitePrep in [1..3];
    case SitePrep of
      1: SSitePrep := 'Mechanical';
      2: SSitePrep := 'Chemical';
      3: SSitePrep := 'None';
    end;   { of case}
  end;   { of set site prep}

Procedure SetSitePrepCost;
  var   x: real;

  begin
    case SitePrep of
              1 : begin
                  clrscr;  gotoxy(5,5);
                  write('You have chosen Mechanical site
                  preparation.');
                  gotoxy(5,7);write('You must choose a cost per acre
                  for this site treatment.');
                  gotoxy(5,8);write('You may enter a value you have
                  in mind, or you may accept the ');
                  gotoxy(5,9);write('default value of
                  ',Mechprep:3:2,' dollars per acre.');
                  gotoxy(5,10);write('To use the default value enter
                  0, otherwise enter your value below.');
                  gotoxy(5,14);write('Site preparation cost per acre
                  is  (xx.xx)   ');
                  range := ['0'..'9','.'];
                  readln(x);
                  if x = 0 then SitePrepCost := Mechprep
                  else SitePrepCost := x;
                  end;

            2 :   begin
                  clrscr;  gotoxy(5,5);
                  write('You have chosen Chemical site
                  preparation.');
                  gotoxy(5,7);write('You must choose a cost per acre
                  for this site treatment.');
                  gotoxy(5,8);write('You may enter a value you have
                  in mind, or you may accept the ');
                  gotoxy(5,9);write('default value of
                   ',Chemprep:3:2,' dollars per acre.');
                  gotoxy(5,10);write('To use the default value enter
                  0, otherwise enter your value below.');
                  gotoxy(5,14);write('Site preparation cost per acre
                  is  (xx.xx)   ');
                  range := ['0'..'9','.'];
                  readln(x);
                  if x = 0 then SitePrepCost := Chemprep
                  else SitePrepCost := x;
                  end;

             3  : SitePrepCost := 0;
    end;  { of case}
 end;      { of set site prep cost}

Procedure SetPlantingMethod;
  begin
    clrscr; gotoxy(5,7);
    write('1.  Machine plant.');
    gotoxy(5,9); write('2.  Hand plant.');
    repeat
    gotoxy(1,12); delline; gotoxy(10,12);
    write('Enter the number of your choice  of planting method.  ');
    range := ['1','2'];
    readln(PlantingMethod);
    until PlantingMethod in [1,2];
    case PlantingMethod of
      1: SPlantingMethod := 'Machine ';
      2: SPlantingMethod := 'Hand ';
```

```
    end;   {of case}
  end;   { of SetPlantingMethod}


Procedure SetPlantingMethodCost;
  var  x: real;

  begin
    if SitePrep = 3 then begin
      if StartTPA < 800 then begin
         MPCost := 134.41;
         HPCost := 124.25;
      end
       else
         MPCost := 146.65;
         HPCost := 116.02;
      end;
    if SitePrep <> 3 then begin
      if StartTPA < 800 then begin
         MPCost := 88.09;
         HPCost := 109.50;
       end
       else
         MPCost := 136.24;
         HPCost := 148.97;
     end;
  case PlantingMethod of
               1 : begin
                   clrscr;  gotoxy(5,5);
                   write('You have elected to use Machine
                   Planting.');
                   gotoxy(5,7);write('You must choose a cost per acre
                   for this activity.');
                   gotoxy(5,8);write('You may enter a value you have
                   in mind, or you may accept the ');
                   gotoxy(5,9);write('default value of ',MPCost:3:2,'
                   dollars per acre.');
                   gotoxy(5,10);write('To use the default value enter
                   0, otherwise enter your value below.');
                   gotoxy(5,14);write('Machine planting cost per acre
                   is  (xx.xx)   ');
                   range := ['0'..'9','.'];
                   readln(x);
                   if x = 0 then PlantingCost := MPCost
                   else PlantingCost := x;
                   end;

              2 :  begin
                   clrscr;  gotoxy(5,5);
                   write('You have elected to use Hand Planting.');
                   gotoxy(5,7);write('You must choose a cost per acre
                   for this activity.');
                   gotoxy(5,8);write('You may enter a value you have
                   in mind, or you may accept the ');
                   gotoxy(5,9);write('default value of ',HPCost:3:2,'
                   dollars per acre.');
                   gotoxy(5,10);write('To use the default value enter
                   0, otherwise enter your value below.');
                   gotoxy(5,14);write('Machine planting cost per acre
                   is  (xx.xx)   ');
                   range := ['0'..'9','.'];
                   readln(x);
                   if x = 0 then PlantingCost := HPCost
                   else PlantingCost := x;
                   end;
      end;    { of case}
    end;   {of SetPlantingMethodCost}


Procedure SetWoodPrice;
   var  y, z: real;
```

```
  begin
  clrscr; gotoxy(5,5);
  write('The default price for red pine pulpwood is $ 12.00 per
  standard cord.');
  gotoxy(5,7); write('Enter the price per cord you want to use, 0
  for the default.  ');
  range := ['0'..'9','.'];
  readln(y);
  if y = 0 then PricePerCord := 12 else PricePerCord := y;
  gotoxy(5,11); write('The default price for red pine sawlogs is $
  70.00 per MBF (Int. 1/4").');
  gotoxy(5,13); write('Enter the price per MBF you want to use, 0
  for the default.  ');
  readln(z);
  if z = 0 then PricePerMBF := 70 else PricePerMBF := z;
  end;      { of Set Wood Price}


Procedure SetInterestRate;
    var y: real;
    begin
      clrscr; gotoxy(5,5);
      write('Select the interest rate you wish to use for
      discounting costs and ');
      gotoxy(5,6); write('revenues.  This is a real rate  (beyond
      inflation).  The default');
      gotoxy(5,7); write('rate is 5 %.');
      gotoxy(10,10); write('Enter the interest rate you want to use,
      0 for the default.  ');
      range := ['0'..'9','.'];
      readln(y);
      if y = 0 then InterestRate := 0.05 else InterestRate := y/100;
      end;    { of SetInterestRate }


Procedure SetLayoutCost;
    var y: real;
    begin
      clrscr; gotoxy(5,5);
      write('You need to choose a sale layout cost per acre.  The
      default value is ');
      gotoxy(5,6); write('$ 9.38 per acre.');
      gotoxy(10,10); write('Enter a value for sale layout costs, 0
      for the default value.  ');
      range := ['0'..'9','.'];
      readln(y);
      if y = 0 then LayoutCost := 9.38 else LayoutCost := y;
      end;      { of SetLayoutCost }


Procedure SetInventoryCost;
   var  y: real;
     begin
     clrscr; gotoxy(5,5);
     write('You need to choose a timber inventory cost per acre.
     The default value');
     gotoxy(5,6); write('is $ 6.73 per acre.');
     gotoxy(10,10);
     write('Enter a value for timber inventory costs, 0 for the
     default.  ');
     range := ['0'..'9','.'];
     readln(y);
     if y = 0 then InventoryCost := 6.73 else InventoryCost := y;
     end;    { of SetInventoryCost }


Procedure DisplaySitePrep;                        {called from
                                              EditSitePrep}

  begin
  clrscr;
```

```
   gotoxy(5,3); writeln('1.  The insect of interest is ',SBugChoice);
   gotoxy(5,5); writeln('2.  Site preparation is going to be
   ',SSitePrep);
   gotoxy(5,7); writeln('3.  Site preparation cost per acre is $
   ',SitePrepCost:3:2);
   gotoxy(5,9); writeln('4.  The trees will be planted by
   ',SPlantingMethod);
   gotoxy(5,11); writeln('5.  Planting cost per acre is $
   ',PlantingCost:3:2);
   gotoxy(5,13); writeln('6.  Annual interest rate is  ',InterestRate
   * 100:3:1,' percent.');
   gotoxy(5,15); writeln('7.  Sale layout cost per acre is $
   ',LayoutCost:3:2);
   gotoxy(5,17); writeln('8.  Inventory cost per acre is $
   ',InventoryCost:3:2);
   gotoxy(5,19); writeln('9.  Price / cord is $  ',PricePercord:3:2,'
   and price / MBF is $  ',PricePerMBF:3:2);
   end;        {of DisplaySitePrep}


Procedure EditSitePrep;                 {called from Basics}
   var       Choice:char;    Which:integer;

   begin
   repeat
   DisplaySitePrep; gotoxy(5,24);
   write('Do you want to change any of these values   (Y or N)?   ');
   range:= ['n','y','N','Y'];
   readln(Choice);
   if upcase(Choice) = #89 then
     begin
     repeat
     gotoxy(1,24); delline; gotoxy(5,24);
     write('Enter the number of the part to change.   ');
     range:= ['1'..'9'];
     readln(Which);
     until (Which>=1) and (Which<=9);
   case Which of
     1: ChooseBug;
     2: SetSitePrep;
     3: SetSitePrepCost;
     4: SetPlantingMethod;
     5: SetPlantingMethodCost;
     6: SetInterestRate;
     7: SetLayoutCost;
     8: SetInventoryCost;
     9: SetWoodPrice;
   end;      {of case}
   end;
until  (Choice = #78) or (Choice = #110);
end;    {of EditSitePrep}


Procedure WaitABit; forward;

Procedure SetSeedlingSurvival;
    var  z : real;

   begin
     z:=0;
     Randomize(0,0);
     SurvivalModifier := random * 0.18 + 0.75;
     if SitePrep = 3 then begin
       z := random * 0.3;
       TPA := round(TPA * (SurvivalModifier - z));
       BugTPA := TPA;
     end
     else
       TPA := round(TPA * SurvivalModifier);
       BugTPA := TPA;
   clrscr;  gotoxy(10,11);
   write('The winds of fate have decreed that your 1st year survival
```

```
  ');
  gotoxy(10,12); write('rate will be ',100 *
  (SurvivalModifier-z):3:1,' %.');
  gotoxy(10,14);
  write('Your plantation now contains ',TPA,' trees / acre.');
  WaitABit;
  end;        {of Set Seedling Survival}


Procedure ThinningOption1;        {called from AThinningType}

  begin
    TreesToRemove := 0;
    clrscr; gotoxy(5,10);
    writeln('You have chosen a thinning treatment which removes a
    specific number of ');
    gotoxy(5,11);writeln('trees per acre.');
    repeat
    gotoxy(1,15); delline; gotoxy(5,15);
    write('How many trees per acre should be removed?   ');
    range:= ['0'..'9'];
    readln(TreesToRemove);
    until (TreesToRemove >=0) and (TreesToRemove <= TPA);
  end;       {of ThinningOption1}


Procedure ThinningOption2;          {called from AThinningType}

  begin
    BAProToRemove := 0;
    clrscr; gotoxy(5,10);
    writeln('You have chosen a thinning treatment which removes a
    constant proportion ');
    gotoxy(5,11);writeln('of the stands basal area or trees per acre
    at each thinning.');
    repeat
    gotoxy(1,15); delline; gotoxy(5,15);
    writeln('What proportion of the stand should be removed?');
    gotoxy(37,16);
    write('(express as a decimal)    ');
    range:= ['0'..'9','.'];
    readln(BAProToRemove);
    until (BAProToRemove >=0) and (BAProToRemove <=1.0);
  end;   {of ThinningOption2}


Procedure ThinningOption3;          {called from AThinningType}

  begin
    BAToLeave :=0;
    clrscr; gotoxy(5,10);
    writeln('You have chosen a thinning treatment which will leave a
    specified ');
    gotoxy(5,11);writeln('residual basal area per acre.');
    repeat
    gotoxy(1,15); delline; gotoxy(5,15);
    write('What should residual basal area per acre be
    (sq.ft./acre)?    ');
    range:= ['0'..'9','.'];
    readln(BAToLeave);
    until (BAToLeave >=0);
    if (StandAge >= MinTAge) and (BAToLeave >= BA) then
      begin
      gotoxy(1,15); delline; gotoxy(15,15);
      writeln('You have chosen to remove the entire stand.  If this
      was an error,'); gotoxy(15,16);
      writeln('you will have the opportunity to change this
      parameter in a minute.');
      delay(4000);
      end;
  end;       {of ThinningOption3}
```

```
Procedure AThinningType;              {called from ThinningOptions}

  begin
    clrscr; gotoxy(5,5);
    writeln('Please choose one of the following thinning regimes:');

    gotoxy(10,8); writeln('1.  Remove a specific number of trees per
    acre.');
    gotoxy(10,10); writeln('2.  Remove a constant proportion of the
    stands basal area, or trees.');
    gotoxy(10,12); writeln('3.  Thin to a residual basal area
    level.');
    gotoxy(1,16); delline; gotoxy(15,16);
    write('Your choice is . . . .    ');
    range:= ['1'..'3'];
    readln(ThinningType);
     case ThinningType of
      1: ThinningOption1;
      2: ThinningOption2;
      3: ThinningOption3;
    end;      {of case}
  end;    {of AThinningType}


Procedure SetMinTAge;               {called from ThinningTiming}

  begin
    clrscr; repeat
    gotoxy(5,1); delline; gotoxy(5,10);
    write('Enter age of first thinning  (years).    ');
    range:= ['0'..'9'];
    readln(MinTAge);
    until MinTAge >= StartAge;
  end;    {of SetMinTAge}


Procedure SetMinTBA;                {called from ThinningTiming}

  begin
    clrscr;gotoxy(5,10);
    writeln('You have the option of stipulating a minimum basal area
    per acre which ');
    gotoxy(5,11);
    writeln('must be present before the first thinning is
    undertaken.  The first ');
    gotoxy(5,12);
    writeln('thinning will not take place until this basal area
    level is reached.');
    gotoxy(5,13); writeln('An entry of 0 will cause thinning to
    begin at the specified min. age.');
    gotoxy(5,16);
    write('Specify a minimum basal area per acre for first thinning.
    ');
    range:= ['0'..'9'];
    readln(MinTBA);
  end;       {of SetMinTBA}


Procedure SetThinningInterval;          {called from ThinningTiming}

  begin
    clrscr; repeat
    gotoxy(1,10); delline; gotoxy(5,10);
    write('Specify the thinning interval you wish to use  (years).
    ');
    range:= ['0'..'9'];
    readln(ThinningInterval);
    until (ThinningInterval >= 1) and (ThinningInterval <120);
  end;      {of SetThinningInterval}
```

```
Procedure SetMaxTAge;              {called from ThinningTiming}

  begin
    clrscr;
    gotoxy(5,7); writeln('Harvest age is set at ',HarvestAge,'
    years.');
    gotoxy(5,9); writeln('Age at first thinning is ',MinTAge,'
    years.');
    gotoxy(5,11); writeln('Thinning interval is ',ThinningInterval,'
    years.');
    repeat
    gotoxy(1,14); delline; gotoxy(5,15);
    write('Specify stand age at final thinning.   ');
    range:= ['0'..'9'];
    readln(MaxTAge);
    until (MaxTAge >= (StartAge + ThinningInterval)) and (MaxTAge <=
     HarvestAge);
  end;              {of SetMaxTAge}


Procedure ThinningTiming;          {called from ThinningOptions}

  begin
    SetMinTAge;
    SetMinTBA;
    SetThinningInterval;
    SetMaxTAge;
  end;        {of ThinningTiming}


Procedure DisplayThinningOptions;      {called from
                                   EditThinningOptions}       begin
  clrscr;
  case ThinningType of
    1: begin
       gotoxy(5,5);
       writeln('1.  Thinning Method:  Remove a specific # of trees /
       acre.');
       gotoxy(5,7);writeln('2.  Number of trees to remove:
        ',TreesToRemove);
       end;
    2: begin
       gotoxy(5,5);
       writeln('1.  Thinning Method:  Remove a constant proportion
       of BA or trees / acre.'); gotoxy(5,7);
       writeln('2.  Proportion of BA or trees to remove:
       ',BAProToRemove:1:3);
       end;
    3: begin
       gotoxy(5,5);
       writeln('1.  Thinning Method:  Constant residual basal area
       per acre.');
       gotoxy(5,7);
       writeln('2.  Residual BA per acre:  ',BAToLeave:3:1);
       end;
  end;     {of case}

    gotoxy(5,9); writeln('3.  Age at first thinning:  ',MinTAge);
    gotoxy(5,11); writeln('4.  Minimum BA / acre before thinning
    begins:  ',MinTBA:3:1);
    gotoxy(5,13); writeln('5.  Thinning interval:
     ',ThinningInterval);
    gotoxy(5,15); writeln('6.  Age at last thinning:  ',MaxTAge);
end;     {of DisplayThinningOptions}


Procedure EditThinningOptions;           {called from
                                      ThinningOptions}
  Var      Choice:char;  Which:integer;

  begin
```

```
    repeat
    DisplayThinningOptions; gotoxy(5,20);
    write('Do you want to change any of these values  (Y or N)?
    ');
    range:= ['y','n','Y','N'];
    readln(Choice);
    if upcase(Choice) = #89 then
      begin
      repeat
      gotoxy(1,20); delline; gotoxy(5,20);
      write('Enter the number of the part to change.    ');
      range:= ['1'..'6'];
      readln(Which);
      until Which in [1..6];
    case Which of
      1: AThinningType;
      2: case ThinningType of
         1: ThinningOption1;
         2: ThinningOption2;
         3: ThinningOption3;
         end;
      3: SetMinTAge;
      4: SetMinTBA;
      5: SetThinningInterval;
      6: SetMaxTAge;
    end;   {of case}
    end;
    until upcase(Choice) = #78;
  end;   {of EditThinningOptions}


Overlay Procedure Spittlebug;

  var
   j, k,SF,OHP,RiskRating,HarvestAgeReducer,HarvestAgeReducer1:
      integer;
   z: real;

  begin
  Randomize(0,0);
  SF := random(101);
  OHP := round(random * (100 - SF) + 0);
  if SF >= 40 then RiskRating := 3;    { 1=low, 2=moderate, 3=high
  risk }
  if OHP > 70 then RiskRating := 3;
  if (OHP >= 0) and (OHP <= 10) then begin
     if (SF >= 30) and (SF <= 35) then RiskRating := 2;
     if (SF > 20) and (SF <= 30) then RiskRating := 1;
     if (SF > 35) and (SF < 40) then RiskRating := 3;
  end;
  if (OHP >= 20) and (OHP <= 30) then begin
     if (SF >= 0) and (SF <= 15) then RiskRating := 1;
     if (SF > 15) and (SF <= 25) then RiskRating := 2;
     if (SF > 25) and (SF <= 30) then RiskRating := 3;
  end;
  if (OHP > 30) and (OHP <= 40) then begin
     if (SF >= 0) and (SF <= 10) then RiskRating := 1;
     if (SF > 10) and (SF <= 20) then RiskRating := 2;
  end;
  if ((OHP >= 0) and (OHP <= 20)) and ((SF >= 0) and (SF <= 20))
     then RiskRating := 1;
  if ((OHP > 10) and (OHP <= 20)) and ((SF > 20) and (SF <= 30))
     then RiskRating := 2;
  if ((OHP > 40) and (OHP <= 70)) and ((SF >= 0) and (SF <= 20))
     then RiskRating := 2;
  if ((OHP > 30) and (OHP <= 70)) and ((SF > 20) and (SF < 40)) then
     RiskRating := 3;
  if ((OHP > 10) and (OHP <= 30)) and ((SF > 30) and (SF < 40)) then
     RiskRating := 3;
  if RiskRating = 1 then SRiskRating := 'Low';
  if RiskRating = 2 then SRiskRating := 'Moderate';
  if RiskRating = 3 then SriskRating := 'High';
```

```
if RiskRating = 1 then      { lose from 1 to 4 years growth }
HarvestAgeReducer := round(random * 4 + 0);
if RiskRating = 2 then      { lose from 4 to 10 years growth }
HarvestAgeReducer := round(random * 6 + 4);
if RiskRating = 3 then begin {lose 10 to 20 prob=.5,else lose all}

z := random;
if z <= 0.5 then HarvestAgeReducer := round(random * 10 + 10)
  else HarvestAgeReducer := HarvestAge;
end;

clrscr; gotoxy(5,5);
write('You have chosen to simulate the effects of Saratoga
spittlebug on');
gotoxy(5,6); write('projected red pine volume yields.  A risk
rating based on randomly');
gotoxy(5,7); write('generated % cover values for sweet-fern and
other alternate host plants');
gotoxy(5,8); write('has been set.  The risk rating is
',SRiskRating,'.  A growth loss of');
gotoxy(5,9); write(HarvestAgeReducer,' years has been randomly set
to correspond to this risk rating.');

gotoxy(5,12); write('There are three insect management
alternatives of interest in this case.');
gotoxy(5,14); write('1. Chemically control alternate host
vegetation before planting.  This');
gotoxy(5,15); write('   will lower the risk rating to low until
the trees are out of danger.');
gotoxy(5,16); write('2. Apply a pesticide to control spittlebug
populations directly.  One');
gotoxy(5,17); write('   spraying for moderate risk sites, two
sprayings for high risk sites, and');
gotoxy(5,18); write('   an extra for either class of site if site
index is <= 50.');
gotoxy(5,19); write('   These treatments will also reduce the risk
rating to low.');
gotoxy(5,20); write('3. Plant the site and ignore the bugs.');
gotoxy(10,24); write('Enter the number of your choice to deal
with this situation.  ');
range := ['1'..'3'];
readln(SpittlebugControlChoice);

if SpittlebugControlChoice = 3 then begin
  if z <= 0.5 then HarvestAge := HarvestAge - HarvestAgeReducer
  else HarvestAge := 0;
  end;
BugHarvestAge := HarvestAge;

if SpittlebugControlChoice = 1 then begin
 if PlantingMethod =2 then SFCCost := 70 else SFCCost := 60;
  clrscr; gotoxy(5,7);
  write('You may enter a cost of alternate host control you have
  in mind or');
  gotoxy(5,8); write('accept the default value of $
  ',SFCCost:2:2,' per acre.');
  gotoxy(10,11); write('Enter the cost (xx.xx), or 0 to use the
  default.  ');
  range := ['0'..'9','.'];
  readln(j);
  if j = 0 then econ[0].control := SFCCost else econ[0].control :=
  j;
  gotoxy(5,15);
  write('A cost for risk rating the plantation must be
  established.  The default');
  gotoxy(5,16); write('value is $ 1.00 per acre.');
  gotoxy(10,19); write('Enter the cost, or 0 to use the default
  value.  ');
  readln(k);
  if k = 0 then econ[0].monitor := 1.00 else econ[0].monitor := k;
  HarvestAgeReducer1 := round(random * 4 + 0);
```

```
    BugHarvestAge := BugHarvestAge - HarvestAgeReducer;
    HarvestAge := HarvestAge - HarvestAgeReducer1;
    if (RiskRating = 2) and ((SI >= 30) and (SI < 70)) then SI:= SI
    + 5;
    if (RiskRating = 3) and ((SI >= 30) and (SI < 70)) then SI:= SI
    + 10;
    end;  { of if controlchoice = 1 then}

  if SpittlebugControlChoice = 2 then begin
    clrscr; gotoxy(5,7);
    write('You may enter a cost of spraying insecticide you have in
    mind or you');
    gotoxy(5,8); write('may accept the default value of $ 12.00 per
    acre.');
    gotoxy(10,11); write('Enter the cost (xx.xx), or 0 to use the
    default.  ');
    range := ['0'..'9','.'];
    SS := 12;
    readln(j);
    if j <> 0 then SS := j;
    gotoxy(5,15);
    write('Spittlebug populations must be monitored in order to
    correctly time spray');
    gotoxy(5,16); write('operations.  For Moderate and High risk
    ratings, monitor 3 times. If site');
    gotoxy(5,17); write('index <= 50 then an extra monitoring must
    be performed.  The default');
    gotoxy(5,18); write('cost for monitoring is $ 1.50 per acre.');
    gotoxy(10,22); write('Enter the cost, or 0 to use the default.
    ');
    SSM := 1.50;
    readln(k);
    if k <> 0 then SSM := k;
    if RiskRating = 1 then begin
     if SI > 50 then begin
      econ[4].monitor := SSM;
      econ[6].monitor := SSM;
      econ[6].control := SS;
      end;
     if SI <= 50 then begin
      econ[4].monitor := SSM;
      econ[6].monitor := SSM;
      econ[6].control := SS;
      econ[10].monitor := SSM;
      end;
    end;
    if RiskRating = 2  then begin
     if SI > 50 then begin
      econ[4].monitor := SSM;
      econ[6].monitor := SSM;
      econ[6].control := SS;
      econ[10].monitor := SSM;
      end;
     if SI <= 50 then begin
      econ[4].monitor := SSM;
      econ[6].monitor := SSM;
      econ[6].control := SS;
      econ[10].monitor := SSM;
      econ[12].monitor := SSM;
      econ[12].control := SS;
     end;
     end;
    if RiskRating = 3 then begin
     if SI > 50 then begin
      econ[4].monitor := SSM;
      econ[8].monitor := SSM;
      econ[4].control := SS;
      econ[9].monitor := SSM;
      econ[9].control := SS;
      end;
     if SI <= 50 then begin
      econ[4].monitor := SSM;
```

```
      econ[8].monitor := SSM;
      econ[4].control := SS;
      econ[9].monitor := SSM;
      econ[9].control := SS;
      econ[14].monitor := SSM;
      econ[14].control := SS;
     end;
     end;
     HarvestAgeReducer1 := round(random * 3 + 0);
     HarvestAge := HarvestAge - HarvestAgeReducer1;
     BugHarvestAge := BugHarvestAge - HarvestAgeReducer;
  end;   { of if control choice = 2}
  end;


Overlay Procedure Grubs;

var   k,j,x,y: real;

  begin
  GCCost := 30;  GMCost := 8;
    Randomize(0,0);
    GrubNumbers := random * 0.75;  {set # of grubs/cu.ft. of soil, 0
                                                            to .75}
    LowGrubMort := 0.24 * GrubNumbers; {set lower limit of
                                                        mortality}
    HighGrubMort := 0.5885714 * Grubnumbers; {set upper limit of
    seedling mort}
    GrubMortPercent := random * (HighGrubMort - LowGrubMort) +
    LowGrubMort;
    SIReducer := round(random * 4 + 5); {generate random SI
    reduction }
    x := random;
    if x <= 0.75 then Year3GrubMort := x * 0.053333
      else Year3GrubMort := 0.32 * x - 0.2;
     y := random;
    if y <= 0.75 then Year4GrubMort := 0
      else Year4GrubMort := 0.08 * y - 0.05;
   clrscr; gotoxy(5,5);
  write('You have chosen to simulate the effects of white grubs on
  projected red ');
  gotoxy(5,6); write('pine volume yields.  A randomly generated
  population of ',Grubnumbers:0:2,' grubs');
  gotoxy(5,7); write('per cubic foot of soil has been established.
  This number may vary between');
  gotoxy(5,8); write('0 and 0.75.');
  gotoxy(5,10); write('A combined 1st and 2nd year seedling
  mortality of ',(GrubMortPercent*100):2:0,' percent has');
  gotoxy(5,11); write('been set to correspond to this population
  level.');
  gotoxy(5,13); write('Seedling mortality in year 3 will be
  ',(Year3GrubMort*100):2:0,' percent.');
  gotoxy(5,14); write('Seedling mortality in year 4 will be
  ',(Year4GrubMort*100):2:0,' percent.');
  gotoxy(5,17); write('There are two insect management alternatives
  of interest in this case.');
  gotoxy(5,20); write('1.  Apply a pesticide at the time of
  planting.');
  gotoxy(5,21); write('2.  Plant the site and ignore the grubs, and
  incur no immediate cost.');
  gotoxy(10,24); write('Enter the number of your grub control
  choice.');
  range := ['1','2'];
  readln(GrubControlChoice);
  if GrubControlChoice = 1 then begin
    clrscr; gotoxy(5,7);
    write('You may enter a cost of grub control you have in mind or
    accept the');
    gotoxy(5,8); write('default value of $',GCCost:2:2,' per
    acre.');
    gotoxy(10,11); write('Enter the cost (xx.xx) or enter 0 to use
    the default.  ');
```

```
    range := ['0'..'9','.'];
    readln(j);
    if j = 0 then GrubControlCost := GCCost else GrubControlCost :=
    j;
    gotoxy(5,15); write('The proposed planting site must be
    monitored before control actions');
    gotoxy(5,16); write('can begin.  The default cost of monitoring
    is $',GMCost:2:2,' per acre.');
    gotoxy(10,19); write('Enter the cost or enter 0 to use the
    default value.  ');
    readln(k);
    if k = 0 then GrubMonitorCost := GMCost else GrubMonitorCost :=
    k;
    TPA := TPA - (round(TPA * ((GrubMortPercent + Year3GrubMort +
     Year4GrubMort) * 0.17)));
    BugTPA := round(( 1 - (GrubMortPercent + Year3GrubMort +
                Year4GrubMort)) * BugTPA);
    BugSI := BugSI - SIReducer;
    econ[0].monitor := GrubMonitorCost;
    econ[0].control := GrubControlCost;
    end;    { of ControlChoice = 1  }
    if GrubControlChoice = 2 then begin                { no control }

    GrubControlCost := 0;
    GrubMonitorCost := 0;
    BugTPA := round(( 1 - (GrubMortPercent + Year3GrubMort +
                Year4GrubMort)) * BugTPA);
    TPA := BugTPA;
    BugSI := BugSI - SIReducer;
    SI := BugSI;
    end;      { of ControlChoice = 2  }
 end;       {of Grubs}


 Overlay Procedure Weevil;

var  ControlChoice,HazardZone,InfestDistance: integer;
     ProbHDam, HDamProb,j,k,g,PruneCost,
     SprayCost: real;

begin
  Randomize(0,0);
  HazardZone := round(random * 2 + 1);
  if HazardZone = 1 then SHazardZone := 'Low';
  if HazardZone = 2 then SHazardZone := 'Medium';
  if HazardZone = 3 then SHazardZone := 'High';

  InfestDistance := random(1101);
  if HazardZone = 1 then begin
  if InfestDistance <= 125 then ProbHDam := 0.05;
   SID := '< 1/8 mile';
 if (InfestDistance >=126) and (InfestDistance <=500) then ProbHDam
 := 0.05;
   SID := '< 1/2 mile';
 if (InfestDistance >=501) and (InfestDistance <=1000) then
 ProbHDam:= 0.05;
   SID := '< 1 mile';
 if InfestDistance > 1000 then ProbHDam := 0;
   SID := '> 1 mile';
end;
if HazardZone = 2 then begin
 if InfestDistance <= 125 then ProbHDam := 0.15;
   SID := '< 1/8 mile';
 if (InfestDistance >=126) and (InfestDistance <=500) then ProbHDam
 := 0.10;
   SID := '< 1/2 mile';
 if (InfestDistance >=501) and (InfestDistance <=1000) then
 ProbHDam:= 0.05;
   SID := '< 1 mile';
 if InfestDistance > 1000 then ProbHDam := 0.05;
   SID := '> 1 mile';
end;
```

```
 if HazardZone = 3 then begin
  if InfestDistance <= 125 then ProbHDam := 0.50;
    SID := '< 1/8 mile';
  if (InfestDistance >=126) and (InfestDistance <=500) then ProbHDam
  := 0.40;
    SID := '< 1/2 mile';
  if (InfestDistance >=501) and (InfestDistance <=1000) then
  ProbHDam:= 0.25;
    SID := '< 1 mile';
  if InfestDistance > 1000 then ProbHDam := 0.10;
    SID := '> 1 mile';
 end;
repeat
 repeat
 HDamProb := random;
 until HDamProb <= ProbHDam;
 if HDamProb <= ProbHDam then begin
     if HazardZone = 1 then WMort := ((random * 5 + 0)/100);
     if HazardZone = 2 then WMort := ((random * 5 + 5)/100);
     if HazardZone = 3 then WMort := ((random * 10 + 5)/100);
  end;  { of HDamProb < ProbHDam }
until WMort >= 0;
if HDamProb > ProbHDam then WMort := 0;

 clrscr; gotoxy(5,5);
 write('You have chosen to simulate the effects of pine root collar
 weevil on');
 gotoxy(5,6); write('projected red pine volume yields.  This
 plantation was first randomly');
 gotoxy(5,7); write('assigned to the ',SHazardZone,' root collar
 weevil hazard zone.  Next');
 gotoxy(5,8); write('a distance of ',SID,' to the nearest weevil
 infestation was ');
 gotoxy(5,9); write('generated.  Using this information red pine
 mortality of ',(WMort*100):2:0,' % due');
 gotoxy(5,10); write('pine root collar weevil was established.');
 gotoxy(5,13); write('There are three insect management alternatives
 of interest in this');
 gotoxy(5,14); write('situation.');
 gotoxy(5,17); write('1. Basal prune and scrape soil and duff away
 from the trees.');
 gotoxy(5,18); write('2. Apply insecticide to the lower few inches of
 each tree trunk.');
 gotoxy(5,19); write('3. Do nothing.');
 gotoxy(5,23); write('Enter the number of your pine root collar
 weevil control choice.  ');
 range := ['1'..'3'];
 readln(ControlChoice);
 if ControlChoice = 1 then begin
   PruneCost := (BugTPA/60) * 6;   {assumes $6.00/hr. & 60
 trees/hour}
   clrscr;
   gotoxy(5,7); write('You may enter a cost per acre for pruning &
   scraping, or accept');
   gotoxy(5,8); write('the default value of $ ',PruneCost:2:2,' per
   acre.');
   gotoxy(10,11); write('Enter the cost (xx.xx), or enter 0 to use
   the default value.  ');
   range := ['0'..'9','.'];
   readln(j);
   if j = 0 then WeevilControlCost := PruneCost else
   WeevilControlCost := j;
   gotoxy(5,15); write('The plantation must be monitored at age 4 and
   age 7.');
   gotoxy(5,16); write('The default cost of monitoring is $ 0.50 per
   tree.');
   gotoxy(10,19); write('Enter the cost of monitoring, or 0 for the
   default.  ');
   readln(k);
   if k = 0 then g:= 0.5 else g:= k;
   if StandArea <= 1 then WeevilMonitorCost := 20 * g;
   if (StandArea > 1) and (StandArea <= 3) then WeevilMonitorCost :=
```

```
  7 * StandArea * g;
  if (StandArea > 3) and (StandArea <=5) then WeevilMonitorCost := 4
  * StandArea * g;
  if (StandArea > 5) and (StandArea <= 10) then WeevilMonitorCost :=
  3 * StandArea * g;
  if StandArea > 10 then WeevilMonitorCost := 2 * StandArea * g;
  econ[7].control := WeevilControlCost;
  econ[3].monitor := WeevilMonitorCost;
  econ[6].monitor := WeevilMonitorCost;
  BugTPA := round((1 - WMort) * BugTPA);
  end;   {of if control choice = 1}
  if ControlChoice = 2 then begin
  SprayCost := ((BugTPA/120) * 6 + 5 );   {assumes $6.00/hr. & 120
  trees/hour and $5/acre for chemicals}
  clrscr;
  gotoxy(5,7); write('You may enter a cost per acre for spraying or
  accept');
  gotoxy(5,8); write('the default value of $ ',SprayCost:2:2,' per
  acre.');
  gotoxy(10,11); write('Enter the cost (xx.xx), or enter 0 to use
  the default value.  ');
  range := ['0'..'9','.'];
  readln(j);
  if j = 0 then WeevilControlCost := SprayCost else
  WeevilControlCost := j;
  gotoxy(5,15); write('The plantation must be monitored at age 4 and
  age 7.');
  gotoxy(5,16); write('The default cost of monitoring is $ 0.50 per
  tree.');
  gotoxy(10,19); write('Enter the cost of monitoring, or 0 for the
  default.  ');
  readln(k);
  if k = 0 then g:= 0.5 else g:= k;
  if StandArea <= 1 then WeevilMonitorCost := 20 * g;
  if (StandArea > 1) and (StandArea <= 3) then WeevilMonitorCost :=
  7 * StandArea * g;
  if (StandArea > 3) and (StandArea <=5) then WeevilMonitorCost := 4
  * StandArea * g;
  if (StandArea > 5) and (StandArea <= 10) then WeevilMonitorCost :=
  3 * StandArea * g;
  if StandArea > 10 then WeevilMonitorCost := 2 * StandArea * g;
  econ[7].control := WeevilControlCost;
  econ[3].monitor := WeevilMonitorCost;
  econ[6].monitor := WeevilMonitorCost;
  BugTPA := round((1 - WMort) * BugTPA);
  end;    { of if control choice = 2}
  if ControlChoice = 3 then begin
  BugTPA := round((1 - WMort) * BugTPA);
  TPA := BugTPA;
  end;  { of if controlchoice = 3 }
end;      {of weevil}


 Overlay Procedure Sawfly;

 var
      controlchoice,j,k,temp1,temp2,temp3: integer;
      SprayCost: real;

  begin
  j:=0; temp1:=0; temp2 :=0; temp3:=0;
Randomize (0,0);
  OutbreakYear := round(random * 7 + 6);
  if BugSI < 50 then begin
    SRC3Percent := random;
    SRC2Percent := (1 - SRC3Percent) * random;
    SRC1Percent := 1 - SRC3Percent - SRC2Percent;
  end;
  if BugSI > 50 then begin
    SRC2Percent := random;
    SRC1Percent := 1 - SRC2Percent;
    SRC3Percent := 0;
```

```
    end;
    Mort1 := random * 0.013 + 0;
    Mort2 := random * 0.018 + 0;
    temp1 := round((SRC1Percent * BugTPA) * (1 - Mort1));
    temp2 := round((SRC2Percent * BugTPA) * (1 - Mort2));
    BugTPA := temp1 + temp2;

clrscr; gotoxy(5,5);
write('You have chosen to simulate the effects of red-headed pine
sawfly on');
gotoxy(5,6); write('projected red pine volume yields.  The year of
sawfly outbreak was ');
gotoxy(5,7); write('randomly determined to be year ',OutbreakYear,'.
The stand has been');
gotoxy(5,8); write('classified as ',(SRC1Percent * 100):3:0,' % SRC
I, ',(SRC2Percent * 100):3:0, ' % SRC II');
gotoxy(5,9); write('and ',(SRC3Percent * 100):3:0,' % SRC III.  SRC
I mortality is ',(Mort1 * 100):2:1,' % and');
gotoxy(5,10); write('SRC II mortality is ',(Mort2 * 100):2:1,' %.
SRC III mortality is 100%.');
gotoxy(5,13); write('There are three insect management alternatives
of interest in this');
gotoxy(5,14); write('situation.');
gotoxy(5,17); write('1. Apply herbicide at planting to control
competing vegetation.');
gotoxy(5,18); write('2. Apply insecticide to the SRC III portion of
the stand.');
gotoxy(5,19); write('3. Do nothing.');
gotoxy(10,23); write('Enter the number of your red-headed pine
sawfly control choice.');
range := ['1'..'3'];
readln(ControlChoice);
if ControlChoice = 1 then begin
j:=0;
    if PlantingMethod = 2 then SFCCost := 70 else SFCCost := 60;
    clrscr;
    gotoxy(5,7); write('You may enter a cost per acre for weed
    control, or accept');
    gotoxy(5,8); write('the default value of $ ',SFCCost:2:2,' per
    acre.');
    gotoxy(10,11); write('Enter the cost (xx.xx), or enter 0 to use
    the default value.  ');
    range := ['0'..'9','.'];
    readln(j);
    if j = 0 then SawflyControlCost := SFCCost else SawflyControlCost
    := j;
    econ[0].control := SawflyControlCost;
    temp3 := round((SRC3Percent * TPA) * (1 - Mort2));
    TPA := temp1 + temp2 + temp3;
    end;   {of if control choice = 1}
if ControlChoice = 2 then begin
j:=0; k:=0; clrscr;
    SprayCost := 12;
    gotoxy(5,7); write('You may enter a cost per acre for spraying or
    accept');
    gotoxy(5,8); write('the default value of $ ',SprayCost:2:2,' per
    acre.');
    gotoxy(10,11); write('Enter the cost (xx.xx), or enter 0 to use
    the default value.  ');
    range := ['0'..'9','.'];
    readln(j);
    if j = 0 then SawflyControlCost := SprayCost else
    SawflyControlCost := j;
    gotoxy(5,15); write('The plantation must be monitored before and
    after spraying.');
    gotoxy(5,16); write('The default cost of monitoring is $ 1.50 per
    acre.');
    gotoxy(10,19); write('Enter the cost of monitoring, or 0 for the
    default.  ');
    readln(k);
    if k = 0 then SawflyMonitorCost := 1.50 else SawflyMonitorCost :=
    k;
```

```
    econ[OutbreakYear].control := SawflyControlCost * (StandArea *
    SRC3Percent);
    econ[OutbreakYear - 1].monitor := SawflyMonitorCost;
    econ[OutbreakYear].monitor := SawflyMonitorCost;
    temp3 := round((SRC3Percent * TPA) * (1 - Mort2));
    TPA := temp1 + temp2 + temp3;
    end;    { of if control choice = 2}
  If ControlChoice = 3 then TPA := BugTPA;
  end;   {of Sawfly}


  Overlay Procedure BugDisplay1;          {called from BugGrow}
    var  x: integer;
    begin
    clrscr;  gotoxy(2,1); write('Stand 1 trees/ac = ',StartTPA1);
    gotoxy(30,1); write('Stand 1 harvest age = ',HarvestAge);
    gotoxy(62,1); write('Stand 1 SI = ',SI);
    gotoxy(2,2); write('Stand 2 trees/ac = ',StartTPA2);
    gotoxy(30,2); write('Stand 2 harvest age = ',BugHarvestAge);
    gotoxy(62,2); write('Stand 2 SI = ',BugSI);
    gotoxy(2,3); write('Minimum pole dbh = ',PoleMin:2:1);
    gotoxy(30,3); write('Minimum sawtimber dbh = ',SawMin:2:1);
    gotoxy(62,3); write(' ');
    for x:= 1 to 80 do begin  gotoxy(x,4);write('-');end;
    end;       {of BugDisplay1}


  Overlay Procedure Display1;          {called from grow}
    var  x: integer;
    begin
    clrscr;  gotoxy(2,2); write('Stand age at start = ',StartAge);
    gotoxy(62,2); write('Site index = ',SI);
    gotoxy(2,3); write('Original BA/ac = ',StartBA:3:1);
    gotoxy(30,3); write('Original trees/ac = ',StartTPA);
    gotoxy(62,3); write('Ave. dbh =
    ',(sqrt((Startba/StartTPA)*183.3466)):2:1);
    gotoxy(2,4); write('Minimum pole dbh = ',PoleMin:2:1);
    gotoxy(30,4); write('Minimum sawtimber dbh = ',SawMin:2:1);
    gotoxy(62,4); write('Harvest age = ',HarvestAge);
    for x:= 1 to 80 do begin  gotoxy(x,5);write('-');end;
    end;       {of Display1}
  Overlay Procedure BugCurrentDisplay;                {called from
                                                          BugGrow}
       var  g,i,z,x,y:integer;   h: real;
    begin
    x:= (StandAge - 1);
    y:= (StandAge - 1);
    if Standage >= BugHarvestAge then y := BugHarvestAge - 1;
    gotoxy(11,5); write('Stand 1: Per Acre Description at the
    Beginning of Year ',x+1);
    gotoxy(9,6);
write('-------------------------------------------------------------');
    gotoxy(1,7); write(SBugChoice);
    gotoxy(25,7); write('Poletimber'); gotoxy(38,7); write('
    Sawtimber');
    gotoxy(59,7); write('  Pole + Saw');
    gotoxy(24,8); write(' ------------  ----------
    -------------------------');
    gotoxy(1,9); write('Dom');  gotoxy(6,9); write('Trees');
    gotoxy(18,9); write('Ave.');  gotoxy(25,9); write('Cu.');
    gotoxy(39,9); write('Cu.'); gotoxy(55,9); write('Cu.');
    gotoxy(1,10); write('Ht.');  gotoxy(7,10); write('/ac.');
    gotoxy(13,10); write('BA'); gotoxy(18,10); write('dbh');
    gotoxy(26,10); write('ft.'); gotoxy(32,10); write('Cords');
    gotoxy(40,10);write('ft.'); gotoxy(47,10); write('MBF');
    gotoxy(56,10); write('ft.'); gotoxy(63,10); write('MAI');
    gotoxy(69,10); write('Cords'); gotoxy(77,10); write('MAI');
    gotoxy(1,11); write('---  -----  ---  ----  -----  ------  ----
    -----  ------------  -----------');
    gotoxy(1,12); write(r[x].DomHt:3:0);
    gotoxy(7,12); write(r[x].TPA);
    gotoxy(13,12); write(r[x].BA:3:0);
```

```
gotoxy(18,12); write(r[x].QMdbh:2:1);
gotoxy(24,12); write(r[x].MerchPoleCFV:4:0,'*');
gotoxy(32,12); write(r[x].PoleCords:3:1,'*');
gotoxy(39,12); write(r[x].MerchSawCFV:4:0,'*');
gotoxy(46,12); write(r[x].MBFVolInt:2:1,'**');
gotoxy(54,12); write(r[x].MerchCFV:5:0,'*');
gotoxy(62,12); write(r[x].MerchCFV/x:3:1);
gotoxy(69,12); write(r[x].MerchCords:2:1,'*');
gotoxy(76,12); write(r[x].MerchCords/x:3:2);
gotoxy(24,13); write(r[x].PoleCFV:4:0,'^');
gotoxy(39,13); write(r[x].SawCFV:4:0,'^');
gotoxy(54,13); write((r[x].PoleCFV + r[x].SawCFV):5:0,'^');
gotoxy(62,13); write((r[x].PoleCFV + r[x].SawCFV)/x:4:1,'^');
gotoxy(1,14); write('Total Non-Insect-Killed Trees =
',MortalityTreeCount);
for z:= 1 to 80 do begin  gotoxy(z,16);write('=');end;
gotoxy(11,17); write('Stand 2: Per Acre Description at the
Beginning of Year ',y+1);
gotoxy(9,18);
write('---------------------------------------------------------
----');
gotoxy(1,19); write(bugr[y].DomHt:3:0);
gotoxy(7,19); write(bugr[y].TPA);
gotoxy(13,19); write(bugr[y].BA:3:0);
gotoxy(18,19); write(bugr[y].QMdbh:2:1);
gotoxy(24,19); write(bugr[y].MerchPoleCFV:4:0,'*');
gotoxy(32,19); write(bugr[y].PoleCords:3:1,'*');
gotoxy(39,19); write(bugr[y].MerchSawCFV:4:0,'*');
gotoxy(46,19); write(bugr[y].MBFVolInt:2:1,'**');
gotoxy(54,19); write(bugr[y].MerchCFV:5:0,'*');
gotoxy(62,19); write(bugr[y].MerchCFV/y:3:1);
gotoxy(69,19); write(bugr[y].MerchCords:2:1,'*');
gotoxy(76,19); write(bugr[y].MerchCords/y:2:2);
gotoxy(24,20); write(bugr[y].PoleCFV:4:0,'^');
gotoxy(39,20); write(bugr[y].SawCFV:4:0,'^');
gotoxy(54,20); write((bugr[y].PoleCFV + bugr[y].SawCFV):5:0,'^');
gotoxy(62,20); write((bugr[y].PoleCFV +
bugr[y].SawCFV)/y:3:1,'^');
gotoxy(1,21); write('Total Non-Insect-Killed Trees =
',BugMortalityTreeCount);
gotoxy(1,23); write('*3 & **6 in. min. top diams. inside bark.');
gotoxy(48,23); write('^ Tot stem vol, no top diam lim.');
end;      {of BugCurrentDisplay}


Overlay Procedure CurrentDisplay;                   {called from Grow}
    var  g,i,z,x:integer;   h: real;
  begin
  x:= (StandAge-1);
  gotoxy(13,7); write('Per Acre Stand Description at the Beginning
  of Year ',x+1);
  gotoxy(13,8);
  write('----------------------------------------------------');
  gotoxy(25,10); write('Poletimber'); gotoxy(38,10); write('
  Sawtimber');
  gotoxy(59,10); write('  Pole + Saw');
  gotoxy(24,11); write(' ------------  ----------
  --------------------------');
  gotoxy(1,12); write('Dom');  gotoxy(6,12); write('Trees');
  gotoxy(18,12); write('Ave.');  gotoxy(25,12); write('Cu.');
  gotoxy(39,12); write('Cu.'); gotoxy(55,12); write('Cu.');
  gotoxy(1,13); write('Ht.');  gotoxy(7,13); write('/ac.');
  gotoxy(13,13); write('BA'); gotoxy(18,13); write('dbh');
  gotoxy(26,13); write('ft.'); gotoxy(32,13); write('Cords');
  gotoxy(40,13);write('ft.'); gotoxy(47,13); write('MBF');
  gotoxy(56,13); write('ft.'); gotoxy(63,13); write('MAI');
  gotoxy(69,13); write('Cords'); gotoxy(77,13); write('MAI');
  gotoxy(1,14); write('---  -----  ---  ----  -----  ------  ----
  -----   -----------   -----------');
  gotoxy(1,15); write(r[x].DomHt:3:0);
  gotoxy(7,15); write(r[x].TPA);
  gotoxy(13,15); write(r[x].BA:3:0);
```

```
   gotoxy(18,15); write(r[x].QMdbh:2:1);
   gotoxy(24,15); write(r[x].MerchPoleCFV:4:0,'*');
   gotoxy(32,15); write(r[x].PoleCords:3:1,'*');
   gotoxy(39,15); write(r[x].MerchSawCFV:4:0,'*');
   gotoxy(46,15); write(r[x].MBFVolInt:2:1,'**');
   gotoxy(54,15); write(r[x].MerchCFV:5:0,'*');
   gotoxy(62,15); write(r[x].MerchCFV/x:3:1);
   gotoxy(69,15); write(r[x].MerchCords:2:1,'*');
   gotoxy(76,15); write(r[x].MerchCords/x:3:2);
   gotoxy(24,16); write(r[x].PoleCFV:4:0,'^');
   gotoxy(39,16); write(r[x].SawCFV:4:0,'^');
   gotoxy(54,16); write((r[x].PoleCFV + r[x].SawCFV):5:0,'^');
   gotoxy(62,16); write((r[x].PoleCFV + r[x].SawCFV)/x:4:1,'^');
   gotoxy(5,18); write('Total Mortality Trees =
   ',MortalityTreeCount);
   gotoxy(4,21); write('*  3 in. min. top diam. inside bark.');
   gotoxy(4,22); write('** 6 in. min. top diam. inside bark.');
   gotoxy(4,23); write('^  Total stem volume, no top diam. limit.');
   end;       {of CurrentDisplay}


 Overlay Procedure BugThinningYearDisplay;         {called from
                                                            BugGrow}
     var  x,y,z:integer;
   begin
   x:= StandAge;
   gotoxy(11,5); write('Stand 1 Per Acre Description at the Beginning
   of Year ',x+1);
   gotoxy(9,6);
   write('-----------------------------------------------------
----');
   gotoxy(24,7); write('Poletimber'); gotoxy(38,7); write('
   Sawtimber');
   gotoxy(59,7); write('  Pole + Saw'); gotoxy(1,7);
   write(SBugChoice);
   gotoxy(24,8); write('-------------  -----------
   -------------------------');
   gotoxy(1,9); write('Dom');  gotoxy(6,9); write('Trees');
   gotoxy(18,9); write('Ave.');  gotoxy(25,9); write('Cu.');
   gotoxy(39,9); write('Cu.'); gotoxy(55,9); write('Cu.');
   gotoxy(1,10); write('Ht.');  gotoxy(7,10); write('/ac.');
   gotoxy(13,10); write('BA'); gotoxy(18,10); write('dbh');
   gotoxy(26,10); write('ft.'); gotoxy(32,10); write('Cords');
   gotoxy(40,10);write('ft.'); gotoxy(47,10); write('MBF');
   gotoxy(56,10); write('ft.'); gotoxy(63,10); write('MAI');
   gotoxy(69,10); write('Cords'); gotoxy(77,10); write('MAI');
   gotoxy(1,11); write('---  -----  ---  ----  -----  ------  ----
    -----  ------------  -----------');
   gotoxy(1,12); write(r[x].DomHt:3:0);
   gotoxy(7,12); write(r[x].TPA);
   gotoxy(13,12); write(r[x].BA:3:0);
   gotoxy(18,12); write(r[x].QMdbh:2:1);
   gotoxy(24,12); write(r[x].MerchPoleCFV:4:0,'*');
   gotoxy(32,12); write(r[x].PoleCords:3:1,'*');
   gotoxy(39,12); write(r[x].MerchSawCFV:4:0,'*');
   gotoxy(46,12); write(r[x].MBFVolInt:2:1,'**');
   gotoxy(54,12); write(r[x].MerchCFV:5:0,'*');
   gotoxy(62,12); write(r[x].MerchCFV/x:3:1);
   gotoxy(69,12); write(r[x].MerchCords:2:1,'*');
   gotoxy(76,12); write(r[x].MerchCords/x:3:2);
   gotoxy(1,13); write('CUT');
   gotoxy(7,13); write(-1 * t[x].ThinnedTPA);
   gotoxy(13,13); write((-1 * t[x].ThinnedBA):3:0);
   gotoxy(18,13); write(r[x].QMdbh:2:1);
   gotoxy(24,13); write((-1 * t[x].TMerchPoleCFV):4:0);
   gotoxy(32,13); write((-1 * t[x].TPoleCords):3:1);
   gotoxy(39,13); write((-1 * t[x].TMerchSawCFV):4:0);
   gotoxy(46,13); write((-1 * t[x].TMBFVolInt):2:2);
   gotoxy(54,13); write((-1 * t[x].TMerchCFV):5:0);
   gotoxy(69,13);  write((-1 * t[x].TMerchCords):2:1);
   gotoxy(1,14); for y := 1 to 80 do begin gotoxy(y,14);
   write('-');end;
```

```
   gotoxy(1,15); write(r[x].DomHt:3:0);
   gotoxy(7,15); write((r[x].TPA - t[x].ThinnedTPA));
   gotoxy(13,15); write((r[x].BA - t[x].ThinnedBA):3:0);
   gotoxy(18,15); write(r[x].QMdbh:2:1);
   gotoxy(24,15); write((r[x].MerchPoleCFV -
   t[x].TMerchPoleCFV):4:0);
   gotoxy(32,15); write((r[x].PoleCords - t[x].TPoleCords):3:1);
   gotoxy(39,15); write((r[x].MerchSawCFV - t[x].TMerchSawCFV):4:0);
   gotoxy(46,15); write((r[x].MBFVolInt - t[x].TMBFVolInt):2:2);
   gotoxy(54,15); write((r[x].MerchCFV - t[x].TMerchCFV):5:0);
   gotoxy(69,15); write((r[x].MerchCords - t[x].TMerchCords):2:1);
   for z:= 1 to 80 do begin  gotoxy(z,17);write('=');end;
   gotoxy(11,18); write('Stand 2 Per Acre Description at the
   Beginning of Year ',x+1);
   gotoxy(9,19);
   write('-----------------------------------------------------------
----');
   gotoxy(1,20); write(bugr[x].DomHt:3:0);
   gotoxy(7,20); write(bugr[x].TPA);
   gotoxy(13,20); write(bugr[x].BA:3:0);
   gotoxy(18,20); write(bugr[x].QMdbh:2:1);
   gotoxy(24,20); write(bugr[x].MerchPoleCFV:4:0,'*');
   gotoxy(32,20); write(bugr[x].PoleCords:3:1,'*');
   gotoxy(39,20); write(bugr[x].MerchSawCFV:4:0,'*');
   gotoxy(46,20); write(bugr[x].MBFVolInt:2:1,'**');
   gotoxy(54,20); write(bugr[x].MerchCFV:5:0,'*');
   gotoxy(62,20); write(bugr[x].MerchCFV/x:3:1);
   gotoxy(69,20); write(bugr[x].MerchCords:2:1,'*');
   gotoxy(76,20); write(bugr[x].MerchCords/x:2:2);
   gotoxy(1,21); write('CUT');
   gotoxy(7,21); write(-1 * bugt[x].ThinnedTPA);
   gotoxy(13,21); write((-1 * bugt[x].ThinnedBA):3:0);
   gotoxy(18,21); write(bugr[x].QMdbh:2:1);
   gotoxy(24,21); write((-1 * bugt[x].TMerchPoleCFV):4:0);
   gotoxy(32,21); write((-1 * bugt[x].TPoleCords):3:1);
   gotoxy(39,21); write((-1 * bugt[x].TMerchSawCFV):4:0);
   gotoxy(46,21); write((-1 * bugt[x].TMBFVolInt):2:2);
   gotoxy(54,21); write((-1 * bugt[x].TMerchCFV):5:0);
   gotoxy(69,21);  write((-1 * bugt[x].TMerchCords):2:1);
   gotoxy(1,22); for y := 1 to 80 do begin gotoxy(y,22);
   write('-');end;
   gotoxy(1,23); write(bugr[x].DomHt:3:0);
   gotoxy(7,23); write((bugr[x].TPA - bugt[x].ThinnedTPA));
   gotoxy(13,23); write((bugr[x].BA - bugt[x].ThinnedBA):3:0);
   gotoxy(18,23); write(bugr[x].QMdbh:2:1);
   gotoxy(24,23); write((bugr[x].MerchPoleCFV -
   bugt[x].TMerchPoleCFV):4:0);
   gotoxy(32,23); write((bugr[x].PoleCords -
   bugt[x].TPoleCords):3:1);
   gotoxy(39,23); write((bugr[x].MerchSawCFV -
   bugt[x].TMerchSawCFV):4:0);
   gotoxy(46,23); write((bugr[x].MBFVolInt -
   bugt[x].TMBFVolInt):2:2);
   gotoxy(54,23); write((bugr[x].MerchCFV - bugt[x].TMerchCFV):5:0);
   gotoxy(69,23); write((bugr[x].MerchCords -
   bugt[x].TMerchCords):2:1);
   end;  { of BugThinningYearDisplay }


 Overlay Procedure ThinningYearDisplay;         {called from Grow}
      var  x,y:integer;
   begin
   x:= StandAge;
   gotoxy(25,7); write('Stand Description  (per acre)');
   gotoxy(25,8);write('----------------------------------');
   gotoxy(38,9); write('Poletimber'); gotoxy(51,9); write('
   Sawtimber');
   gotoxy(65,9); write('  All Trees');
   gotoxy(37,10); write(' -----------   ----------    -----------');
   gotoxy(4,11); write('Stand');  gotoxy(11,11); write('Dom');
   gotoxy(17,11); write('Trees');  gotoxy(30,11); write('Ave.');
   gotoxy(38,11); write('Cu.'); gotoxy(52,11); write('Cu.');
```

```
gotoxy(66,11); write('Cu.');
gotoxy(4,12); write('Age');  gotoxy(11,12); write('Ht.');
gotoxy(17,12); write('/ac.'); gotoxy(24,12); write('BA');
gotoxy(30,12); write('dbh'); gotoxy(39,12); write('ft.#');
gotoxy(44,12);write('Cords#'); gotoxy(53,12); write('ft.#');
gotoxy(58,12); write('MBF##'); gotoxy(67,12); write('ft.#');
gotoxy(72,12); write('Cords#');
gotoxy(4,13); write('-----  ---   -----  ---   ---    -----  -----
----  ----    ----  -----');
gotoxy(1,14); write('*');
gotoxy(4,14); write(r[x].StandAge);  gotoxy(11,14);
write(r[x].DomHt:3:0);
gotoxy(17,14); write(r[x].TPA);  gotoxy(24,14);
write(r[x].BA:3:0);
gotoxy(30,14); write(r[x].QMdbh:2:1); gotoxy(37,14);
write(r[x].MerchPoleCFV:4:0); gotoxy(44,14);
write(r[x].PoleCords:3:1);
gotoxy(51,14); write(r[x].MerchSawCFV:4:0);  gotoxy(58,14);
write(r[x].MBFVolInt:2:2); gotoxy(65,14);
write(r[x].MerchCFV:5:0);
gotoxy(72,14);  write(r[x].MerchCords:2:1);
gotoxy(1,15); write('CUT ---------');
gotoxy(17,15); write(-1 * t[x].ThinnedTPA);
gotoxy(23,15); write((-1 * t[x].ThinnedBA):3:0);
gotoxy(30,15); write(r[x].QMdbh:2:1); gotoxy(36,15);
write((-1 * t[x].TMerchPoleCFV):4:0);
gotoxy(43,15); write((-1 * t[x].TPoleCords):3:1);
gotoxy(51,15); write((-1 * t[x].TMerchSawCFV):4:0);
gotoxy(57,15);
write((-1 * t[x].TMBFVolInt):2:2); gotoxy(65,15);
write((-1 * t[x].TMerchCFV):5:0);
gotoxy(71,15);  write((-1 * t[x].TMerchCords):2:1);
gotoxy(4,16); for y := 1 to 80 do begin gotoxy(y,16);
write('-');end;
gotoxy(1,17); write('**');
gotoxy(4,17); write(r[x].StandAge);  gotoxy(11,17);
write(r[x].DomHt:3:0);
gotoxy(17,17); write((r[x].TPA - t[x].ThinnedTPA));
gotoxy(24,17); write((r[x].BA - t[x].ThinnedBA):3:0);
gotoxy(30,17); write(r[x].QMdbh:2:1);
gotoxy(37,17); write((r[x].MerchPoleCFV -
t[x].TMerchPoleCFV):4:0);
gotoxy(45,17); write((r[x].PoleCords - t[x].TPoleCords):3:1);
gotoxy(52,17); write((r[x].MerchSawCFV - t[x].TMerchSawCFV):4:0);
gotoxy(59,17); write((r[x].MBFVolInt - t[x].TMBFVolInt):2:2);
gotoxy(65,17); write((r[x].MerchCFV - t[x].TMerchCFV):5:0);
gotoxy(73,17); write((r[x].MerchCords - t[x].TMerchCords):2:1);
gotoxy(1,20); write('*   Stand Before Thinning.');
gotoxy(35,20); write('**  Stand After Thinning.');
gotoxy(1,22); write('#  3 in. min. top diam. inside bark.');
gotoxy(40,22); write('##  6 in. min. top diam. inside bark.');
end;    { of ThinningYearDisplay }


Procedure CompInt(value:real; age:integer); Forward;


Overlay Procedure NPV;
   var      x: integer;
     temp2,temp3,temp4,temp5,cp,sp,pp,dcp,dsp,dpp,sv,cv,pv,dsv,
     dcv,dpv: real;

   begin

   { calculate discounted costs for management activities }
    temp2:=0; temp3:=0; temp4:=0; temp5:=0;
     for x := 1 to HarvestAge do begin
       with econ[x] do begin
         if monitor <> 0 then begin
           CompInt(monitor,x);
           monitor := DiscountedValue;
```

```
        temp2 := temp2 + monitor;
      end;   { of monitor }
      if control <> 0 then begin
        CompInt(control,x);
        control := DiscountedValue;
        temp3 := temp3 + control;
      end;   { of control }
      if inventory <> 0 then begin
        CompInt(inventory,x);
        inventory := DiscountedValue;
        temp4 := temp4 + inventory;
      end;   { of inventory}
      if layout <> 0 then begin
        CompInt(layout,x);
        layout := DiscountedValue;
        temp5 := temp5 + layout;
      end;   { of layout }
    end;   { of with - do }
  end;   {of for - do }
temp2 := temp2 + econ[0].monitor;
temp3 := temp3 + econ[0].control;

{calculate discounted revenues for controlled stand}
cp:=0; sp:=0; pp:=0; dcp:=0; dsp:=0; dpp:=0;
for x := 1 to HarvestAge do begin
  with t[x] do begin
    if thinnedBA <> 0 then begin
    cp := TMerchCords * PricePerCord;
    sp := TMBFVolInt * PricePerMBF;
    pp := TPoleCords * PricePerCord;
    CompInt(cp,x);
    dcp := dcp + DiscountedValue;
    CompInt(sp,x);
    dsp := dsp + DiscountedValue;
    CompInt(pp,x);
    dpp := dpp + DiscountedValue;
    end;  { of if - then }
  end;  { of with - do }
end;  { of for - do }

sv := r[HarvestAge-1].MBFVolInt * PricePerMBF;
CompInt(sv,HarvestAge);
dsv := DiscountedValue;
cv := r[HarvestAge-1].MerchCords * PricePerCord;
CompInt(cv,HarvestAge);
dcv := DiscountedValue;
pv := r[HarvestAge-1].PoleCords * PricePerCord;
CompInt(pv,HarvestAge);
dpv := DiscountedValue;

DiscountedCordsRevenue := dcp + dcv;
DiscountedSawRevenue := dsp + dpp + dsv + dpv;
AllCostsDiscounted := SitePrepCost + PlantingCost + temp2 +
  temp3 + temp4 + temp5;
InsectControlCostsDiscounted := temp2 + temp3;
 { calculate discounted revenues for bug stand }
cp:=0; sp:=0; pp:=0; dcp:=0; dsp:=0; dpp:=0;
for x := 1 to BugHarvestAge do begin
  with bugt[x] do begin
    if thinnedBA <> 0 then begin
    cp := TMerchCords * PricePerCord;
    sp := TMBFVolInt * PricePerMBF;
    pp := TPoleCords * PricePerCord;
    CompInt(cp,x);
    dcp := dcp + DiscountedValue;
    CompInt(sp,x);
    dsp := dsp + DiscountedValue;
    CompInt(pp,x);
    dpp := dpp + DiscountedValue;
    end;  { of if - then }
  end;  { of with - do }
end;  { of for - do }
```

```
    sv := bugr[BugHarvestAge-1].MBFVolInt * PricePerMBF;
    CompInt(sv,BugHarvestAge);
    dsv := DiscountedValue;
    cv := bugr[BugHarvestAge-1].MerchCords * PricePerCord;
    CompInt(cv,BugHarvestAge);
    dcv := DiscountedValue;
    pv := bugr[BugHarvestAge-1].PoleCords * PricePerCord;
    CompInt(pv,BugHarvestAge);
    dpv := DiscountedValue;

    BugDiscountedCordsRevenue := dcp + dcv;
    BugDiscountedSawRevenue := dsp + dpp + dsv + dpv;

end;   { of NPV }


Overlay Procedure VolumeSummary;
  var   x,y: integer;

  begin
    TTMerchCords:=0; TTMBFVolInt:=0; TTPoleCords:=0; TTTotalCFV:=0;
    BugTTMerchCords:=0; BugTTMBFVolInt:=0; BugTTPoleCords:=0;
     BugTTTotalCFV:=0;
    for x := 1 to HarvestAge do begin
      with t[x] do begin
        if ThinnedBA <> 0 then begin
      TTMerchCords := TTMerchCords + TMerchCords;
      TTMBFVolInt := TTMBFVolInt + TMBFVolInt;
      TTPoleCords := TTPoleCords + TPoleCords;
      TTTotalCFV := TTTotalCFV + TMerchPoleCFV + TMerchSawCFV;
      TThinnedTPA := TThinnedTPA + ThinnedTPA;
      end;  { of if - then }
      end;  { of with - do }
    end;  { of for - do }
    for y := 1 to BugHarvestAge do begin
      with bugt[y] do begin
        if BugThinnedBA <> 0 then begin
      BugTTMerchCords := BugTTMerchCords + TMerchCords;
      BugTTMBFVolInt := BugTTMBFVolInt + TMBFVolInt;
      BugTTPoleCords := BugTTPoleCords + TPoleCords;
      BugTTTotalCFV := BugTTTotalCFV + TMerchPoleCFV + TMerchSawCFV;
      BugTThinnedTPA := BugTThinnedTPA + ThinnedTPA;
      end;  { of if - then }
      end;  { of with - do }
    end;  { of for - do }
  end;  { of VolumeSummary }


Overlay Procedure DisplayVolumeSummary;
   var  x: integer;

  begin
  clrscr;
  x:=HarvestAge-1;
  if Choice1 = 1 then begin
  gotoxy(30,1); write('Stand Volume Summary');
  gotoxy(28,2); write('------------------------');
  end;
  if Choice1 = 2 then begin
  gotoxy(16,1);
  write('Stand 1 Volume Summary (with insect management)');
  gotoxy(14,2);
  write('-----------------------------------------------------');
  end;
  gotoxy(5,4); write('Harvested merchantable cordwood volume = ');
  gotoxy(59,4); write(r[x].MerchCords:4:1,' cords/acre.');
  gotoxy(5,5); write('Merchantable cordwood volume removed in
  thinnings = ');
  gotoxy(59,5); write(TTMerchCords:4:1,' cords/acre.');
  gotoxy(5,6); write('Total merchantable cordwood volume produced =
  ');
```

```
     gotoxy(59,6); write((r[x].MerchCords + TTMerchCords):4:1,'
     cords/acre.');
     gotoxy(5,7); write('MAI for merchantable cordwood volume = ');
     gotoxy(59,7);
     write(((r[x].MerchCords+TTMerchCords)/HarvestAge):2:2,'
     cords/acre/year.');
     gotoxy(5,9); write('Harvested International 1/4" bd. ft. volume =
      ');
     gotoxy(59,9); write(r[x].MBFVolInt:3:1,' MBF/acre.');
     gotoxy(5,10); write('Int. 1/4" bd. ft. volume removed in all
     thinnings = ');
     gotoxy(59,10); write(TTMBFVolInt:3:1,' MBF/acre.');
     gotoxy(5,11); write('Total International 1/4" bd. ft. volume
     produced = ');
     gotoxy(59,11); write((r[x].MBFVolInt+TTMBFVolInt):3:1,'
     MBF/acre.');
     gotoxy(5,12); write('MAI for International 1/4" bd. ft. volume =
      ');
     gotoxy(59,12); write((r[x].MBFVolInt+TTMBFVolInt)/HarvestAge:2:2,'
     MBF/acre/year.');
     gotoxy(5,14); write('Cordwood poles harvested in addition to
     sawtimber = ');
     gotoxy(59,14); write(r[x].PoleCords:3:1,' cords/acre.');
     gotoxy(5,15); write('Cordwood poles removed in all thinnings = ');
     gotoxy(59,15); write(TTPoleCords:3:1,' cords/acre.');
     gotoxy(5,16); write('Total pole volume produced in addition to
     sawtimber = ');
     gotoxy(59,16); write((r[x].PoleCords+TTPoleCords):4:1,'
     cords/acre.');
     gotoxy(5,18); write('Harvested merchantable cubic foot volume =
      ');
     gotoxy(59,18); write((r[x].MerchPoleCFV + r[x].MerchSawCFV):5:0,'
     cu ft/acre.');
     gotoxy(5,19); write('Merchantable cubic foot volume removed in
     thinnings = ');
     gotoxy(59,19); write(TTTotalCFV:5:0,' cu ft/acre.');
     gotoxy(5,20); write('Total merchantable cubic foot volume produced
     = ');
     gotoxy(59,20);
     write((r[x].MerchPoleCFV+r[x].MerchSawCFV+TTTotalCFV):5:0,' cu
     ft/acre.');
     gotoxy(5,21); write('MAI for merchantable cubic foot volume = ');
     gotoxy(59,21);
     write(((r[x].MerchPoleCFV+r[x].MerchSawCFV+TTTotalCFV)/Har
vestAge):4:1,' cu ft/acre/year.');
     end;  { of DisplayVolumeSummary }


   Overlay Procedure DisplayBugVolumeSummary;
      var  x: integer;

     begin
     clrscr; gotoxy(15,1);
     x:=BugHarvestAge-1;
     write('Stand 2 Volume Summary (without insect management)');
     gotoxy(13,2);
     write('----------------------------------------------------');
     gotoxy(5,4); write('Harvested merchantable cordwood volume = ');
     gotoxy(59,4); write(bugr[x].MerchCords:4:1,' cords/acre.');
     gotoxy(5,5); write('Merchantable cordwood volume removed in
     thinnings = ');
     gotoxy(59,5); write(BugTTMerchCords:4:1,' cords/acre.');
     gotoxy(5,6); write('Total merchantable cordwood volume produced =
      ');
     gotoxy(59,6); write((bugr[x].MerchCords + BugTTMerchCords):4:1,'
     cords/acre.');
     gotoxy(5,7); write('MAI for merchantable cordwood volume = ');
     gotoxy(59,7); write(((bugr[x].MerchCords +
     BugTTMerchCords)/HarvestAge):2:2,' cords/acre/year.');
     gotoxy(5,9); write('Harvested International 1/4" bd. ft. volume =
      ');
     gotoxy(59,9); write(bugr[x].MBFVolInt:3:1,' MBF/acre.');
```

```
gotoxy(5,10); write('Int. 1/4" bd. ft. volume removed in all
thinnings = ');
gotoxy(59,10); write(BugTTMBFVolInt:3:1,' MBF/acre.');
gotoxy(5,11); write('Total International 1/4" bd. ft. volume
produced = ');
gotoxy(59,11); write((bugr[x].MBFVolInt + BugTTMBFVolInt):3:1,'
MBF/acre.');
gotoxy(5,12); write('MAI for International 1/4" bd. ft. volume =
');
gotoxy(59,12); write((bugr[x].MBFVolInt +
BugTTMBFVolInt)/HarvestAge:2:2,' MBF/acre/year.');
gotoxy(5,14); write('Cordwood poles harvested in addition to
sawtimber = ');
gotoxy(59,14); write(bugr[x].PoleCords:3:1,' cords/acre.');
gotoxy(5,15); write('Cordwood poles removed in all thinnings = ');
gotoxy(59,15); write(BugTTPoleCords:3:1,' cords/acre.');
gotoxy(5,16); write('Total pole volume produced in addition to
sawtimber = ');
gotoxy(59,16); write((bugr[x].PoleCords + BugTTPoleCords):4:1,'
cords/acre.');
gotoxy(5,18); write('Harvested merchantable cubic foot volume =
');
gotoxy(59,18); write((bugr[x].MerchPoleCFV +
bugr[x].MerchSawCFV):5:0,' cu ft/acre.');
gotoxy(5,19); write('Merchantable cubic foot volume removed in
thinnings = ');
gotoxy(59,19); write(BugTTTotalCFV:5:0,' cu ft/acre.');
gotoxy(5,20); write('Total merchantable cubic foot volume produced
= ');
gotoxy(59,20); write((bugr[x].MerchPoleCFV + bugr[x].MerchSawCFV +
BugTTTotalCFV):5:0,' cu ft/acre.');
gotoxy(5,21); write('MAI for merchantable cubic foot volume = ');
gotoxy(59,21); write(((bugr[x].MerchPoleCFV + bugr[x].MerchSawCFV
+ BugTTTotalCFV)/HarvestAge):4:1,' cu ft/acre/year.');
end;        { of DisplayBugVolumeSummary }


Procedure DisplayEcon;
  var     y: real;
  begin
  clrscr;
  y:= InsectControlCostsDiscounted;
    gotoxy(28,2); write('Final Economic Analysis');
    gotoxy(26,3); write('---------------------------');
    gotoxy(5,5); write('Insect Managed Stand');
    gotoxy(5,6); write('--------------------');
    gotoxy(5,8); write('Discounted value of total cordwood volume =
');
    gotoxy(65,8); write('$ ',DiscountedCordsRevenue:5:2);
    gotoxy(5,9); write('Discounted value of sawtimber plus pole
    cordwood volume = ');
    gotoxy(65,9); write('$ ',DiscountedSawRevenue:5:2);
    gotoxy(5,10); write('Discounted value of all costs = ');
    gotoxy(65,10); write('$ ',AllCostsDiscounted:5:2);
    gotoxy(5,12); write('Net present value of the stand in year of
    planting (year 0) is:');
    gotoxy(10,13); write('Cordwood NPV = $ ',(DiscountedCordsRevenue
    - AllCostsDiscounted):3:2);
    gotoxy(40,13); write('Sawtimber NPV = $ ',(DiscountedSawRevenue
    - AllCostsDiscounted):3:2);
    gotoxy(5,15); write('Evaluation of Insect Management Efforts');
    gotoxy(5,16); write('---------------------------------------');
    gotoxy(5,17); write('Discounted value of cordwood volume saved =
    ');
    gotoxy(65,17); write('$ ',(DiscountedCordsRevenue -
    BugDiscountedCordsRevenue):5:2);
    gotoxy(5,18); write('Discounted value of sawtimber volume saved
    = ');
    gotoxy(65,18); write('$ ',(DiscountedSawRevenue -
    BugDiscountedSawRevenue):5:2);
    gotoxy(5,19); write('Discounted value of insect management costs
    = ');
```

```
        gotoxy(65,19); write('$ ',InsectControlCostsDiscounted:5:2);
        gotoxy(5,21); write('Discounted value of insect management
        efforts :');
        gotoxy(10,22); write('Cordwood NPV = $
        ',((DiscountedCordsRevenue - BugDiscountedCordsRevenue)-y):3:2);
        gotoxy(40,22); write('Sawtimber NPV = $ ',((DiscountedSawRevenue
        - BugDiscountedSawRevenue)-y):3:2);
        end;     { of DisplayEcon }


Program  RPPest;

{$I  vardec.inc     Variable, type, and constant declarations.}
{$I  randomize.inc  }
{$I  inp.inc   }
{$I  initial.inc    Setting initial stand conditions.}
{$I  bugs.inc    }
{$I  Display2.inc  }
{$I  CD2.inc   }
{$I  TYD.inc   }
{$I  final.inc  }

 Procedure WaitABit;
         var   ch: char;
     begin
     gotoxy(24,25);  write('Press the Space Bar to continue.');
     range := [#13,#32];
     repeat
     read(kbd,ch);
     until ch in [#13,#32];
     delline;
     end;

   Procedure AvBA1;
     begin
       AveBA := BA / TPA
     end;

   Procedure BugAvBA;
     begin
       BugAveBA := BugBA / BugTPA;
     end;


   Procedure QMD1;
     begin
       AvBA1;
       QMdbh := sqrt(AveBA * 183.346626);
     end;

   Procedure BugQMD;
     begin
       BugAvBA;
       BugQMdbh := sqrt(BugAveBA * 183.346626);
     end;


   Procedure MDI1;
     begin
       MaxDiamIncr := 0.007 * SI * exp(-0.01 * BHA); {Lundgren, 1981}
     end;

   Procedure BugMDI;
     begin
       BugMaxDiamIncr := 0.007 * BugSI * exp(-0.01 * BugBHA);
                                                    {Lundgren, 1981}
     end;


   Procedure DGRate;
     begin
       DGR := 0;
```

```
      if StandAge <> StartAge then begin
      DGR := QMDbh  - r[StandAge - 1].QMdbh;
      if r[StandAge - 1].QMdbh = 0 then DGR := MaxDiamIncr;
      end
      else DGR := MaxDiamIncr;
      if DGR <= 0 then DGR := 0.0001;
    end;         {of DGRate}


Procedure BugDGRate;
    begin
      BugDGR := 0;
      if StandAge <> StartAge then begin
      BugDGR := BugQMDbh  - Bugr[StandAge - 1].QMdbh;
      if Bugr[StandAge - 1].QMdbh = 0 then BugDGR := BugMaxDiamIncr;
      end
      else BugDGR := BugMaxDiamIncr;
      if BugDGR <= 0 then BugDGR := 0.0001;
    end;        {of BugDGRate}


  Procedure MBAI1;
    begin
      MaxBAIncr := 0.00545415 * TPA * (2 * QMdbh * MaxDiamIncr +
         sqr(MaxDiamIncr));
    end;

  Procedure BugMBAI;
    begin
      BugMaxBAIncr := 0.00545415 * BugTPA * (2 * BugQMdbh *
          BugMaxDiamIncr + sqr(BugMaxDiamIncr));
    end;

  Procedure DH1;
    begin
      DomHt := int(1.89 * SI * (exp(1.3892 * ln(1 - exp(-0.01979 *
      SeedAge)))));
    end;

  Procedure BugDH;
    begin
      BugDomHt := int(1.89 * BugSI * (exp(1.3892 * ln(1 - exp(-0.01979
          * SeedAge)))));
    end;

  Procedure DSD1;
    begin
      DiamSD := 0.37628 * QMdbh * (exp((-0.093346 * QMdbh)));
    end;

  Procedure BugDSD;
    begin
      BugDiamSD :=  0.37628 * BugQMdbh * (exp((-0.093346 *
          BugQMdbh)));
    end;


  Procedure ConvertBFtoCF1;
    begin
      BFperCF :=  -8.76 + (1.985 * QMdbh) - (0.07253 * sqr(QMdbh)) +
        (0.0008421 * exp(3 * ln(QMdbh))) + (0.04951 * DomHt) -
        (0.00892 * QMdbh * DomHt) + (0.0003169 * DomHt * exp(2 *
        ln(QMdbh))) - (0.000002786 * DomHt * exp(3 * ln(QMdbh)));
     if BFperCF <0 then BFperCF := 0;
    end;    {of ConvertBFtoCF1}

  Procedure BugConvertBFtoCF;
    begin
      BugBFperCF :=  -8.76 + (1.985 * BugQMdbh) - (0.07253 *
        sqr(BugQMdbh)) + (0.0008421 * exp(3 * ln(BugQMdbh))) +
        (0.04951 * BugDomHt) - (0.00892 * BugQMdbh * BugDomHt) +
        (0.0003169 * BugDomHt * exp(2 * ln(BugQMdbh))) - (0.000002786
```

```
    * BugDomHt * exp(3 * ln(BugQMdbh)));
  if BugBFperCF <0 then BugBFperCF := 0;
 end;     {of BugConvertBFtoCF}


Procedure NumbersPerClass (ATPA:integer; AQMdbh:real;
                       ADiamSD:real); {called from NumbersPerClass1}
 VAR
    aul,all,sduavel,sduaveu,ztl,ztu,avelz,aveuz,lll,
    uul,sdulll,sduuul,ztlll,ztuul,lllz,lulz,ullz,
    uulz: real;
    c1,c2,x,temp1,temp2,temp3,temp4 : integer;

   begin
     FillChar(A,SizeOf(A),0);
     temp1 :=0; temp2 :=0; temp3 :=0; temp4 := 0;
     if frac(AQMDBH) = 0.50000 then AveClass := round(AQMDBH) - 1
     else AveClass := round(AQMdbh);
     AUL := AveClass + 0.5;              {define upper limit of dbh
                                                          class}
     ALL := AveClass - 0.49999999;    {p.73  Avery & Burkhart,
                                                          1983}
     SDUAveL := abs((ALL - AQMdbh) / ADiamSD);
     SDUAveU := abs((AUL - AQMdbh) / ADiamSD);
     ZTL := 1 / (1 + ZP * SDUAveL);
     ZTU := 1 / (1 + ZP * SDUAveU);
     AveLZ := 0.5 - ((1 / sqrt(2 * pi)) * (exp(-sqr(SDUAveL) / 2))
        * ztl *(zb1 + ZTL * (ZB2 + ZTL * (ZB3 + ZTL * (ZB4 + ZTL *
        ZB5)))));
     AveUZ := 0.5 - ((1 / sqrt(2 * pi)) * (exp(-sqr(SDUAveU) / 2))
        * ZTU *(zb1 + ztu * (ZB2 + ZTU * (ZB3 + ZTU * (ZB4 + ZTU *
        ZB5)))));
     A[AveClass] := round((AveLZ + AveUZ) * ATPA);
     LULZ := AveLZ;
     ULLZ := AveUZ;

     x := 0;
       repeat                         {calculate NOT/diam class}
         x := x + 1;                    {& assign them to array
                                                        Class}
         LowClass := AveClass - x;
         UpClass := AveClass + x;
         LLL := LowClass - 0.49999999;
         UUL := UpClass + 0.5;
         SDULLL := abs((LLL - AQMdbh) / ADiamSD);
         SDUUUL := abs((UUL - AQMdbh) / ADiamSD);
         ZTLLL := 1 / (1 + ZP * SDULLL);
         ZTUUL := 1 / (1 + ZP * SDUUUL);
   LLLZ := 0.5 - ((1/sqrt(2*pi)) * (exp(-sqr(SDULLL)/2)) * ZTLLL *
       (zb1 + ztlll * (ZB2 + ZTLLL * (ZB3 + ZTLLL * (ZB4 + ZTLLL *
       ZB5)))));
   UULZ := 0.5 - ((1/sqrt(2*pi)) * (exp(-sqr(SDUUUL)/2)) * ZTUUL *
       (zb1 + ztuul * (ZB2 + ZTUUL * (ZB3 + ZTUUL * (ZB4 + ZTUUL *
       ZB5)))));

         A[LowClass] := round((LLLZ - LULZ) * ATPA);
         A[UpClass] := round((UULZ - ULLZ) * ATPA);
         if A[LowClass] <> 0 then temp1 := LowClass;
         if A[UpClass] <> 0 then temp2 := Upclass;
         temp3 := temp3 + A[lowclass] + A[upclass];
         LULZ := LLLZ;
         ULLZ := UULZ;
       until (A[LowClass] = 0) and (A[Upclass] = 0);
       temp4 := temp3 + A[aveclass];
       if temp4 < ATPA then A[AveClass] := A[AveClass] + (ATPA -
          Temp4) else
        A[AveClass] := A[AveClass] - (temp4 - ATPA);

        SmallestClass := temp1;
        BugSmallestClass := temp1;
        BiggestClass := temp2;
        BugBiggestClass := temp2;
```

```
    end;        {of NumbersPerClass}


  Procedure NumbersPerClass1;          {called from Grow}
    var y: integer;
    begin
      NumbersPerClass(TPA,QMdbh,DiamSD);
      for y:=SmallestClass to BiggestClass do Class[y] := A[y];
    end;      {of NumbersPerClass1}


  Procedure BugNumbersPerClass;          {called from Grow}
    var y: integer;
    begin
      NumbersPerClass(BugTPA,BugQMdbh,BugDiamSD);
      for y:=BugSmallestClass to BugBiggestClass do BugClass[y] :=
      A[y];
    end;      {of BugNumbersPerClass}


  Procedure TNumbersPerClass;
    var y:integer;
    begin
      NumbersPerClass(ThinnedTPA,QMdbh,DiamSD);
      for y:=SmallestClass to BiggestClass do ThinnedClass[y]:= A[y];
    end;      {of TNumbersPerClass}


  Procedure BugTNumbersPerClass;
    var y:integer;
    begin
      NumbersPerClass(BugThinnedTPA,BugQMdbh,BugDiamSD);
      for y:=BugSmallestClass to BugBiggestClass do
      BugThinnedClass[y]:= A[y];
    end;      {of BugTNumbersPerClass}



      Procedure SawPoleBA (BClass:p2; BTPA:integer; BBA:real;
                            ASmallestClass: integer;
ABiggestClass:integer);
      Var
          z: integer;   temp4,Q: real;

        begin
          SawClass := 0;  PoleClass := 0;
          FillChar(BB,SizeOf(BB),0);
          if frac(PoleMin) = 0.50000 then PoleClass := round(PoleMin) -
            1 else
          PoleClass := round(PoleMin);
          if frac(SawMin) = 0.50000 then SawClass := round(SawMin) - 1
            else
          SawClass := round(SawMin);

          temp4 := 0;
          for z := ASmallestClass to ABiggestClass do begin
               BB[z] := BClass[z] * (sqr(z) * 0.00545415);
               temp4 := temp4 + BB[z];
          end;

          if temp4 <> BBA then begin  {modify BA/class array so it = BA}

            for z := ASmallestClass to ABiggestClass do begin
            Q := (abs(temp4 - BBA) * (BB[z]/temp4));
            BB[z] := BB[z] - Q;
            end;
          end;

          z := 0; CPoleBA := 0;
          for z := PoleClass to (SawClass - 1)  do CPoleBA := CPoleBA +
                   BB[z];
```

```
        CSawBA := 0;  z := 0;
        for z := SawClass to ABiggestClass  do CSawBA := CSawBA +
                 BB[z];

        z := 0; CSmallBA := 0;
        for z := 4 to (PoleClass - 1)  do CSmallBA := CSmallBA +
                BB[z];

      end;        {of SawPoleBA}


  Procedure SawPoleBA1;              {called from Grow}
    var y:integer;
    begin
      SawPoleBA(Class,TPA,BA,SmallestClass,BiggestClass);
      for y:= SmallestClass to BiggestClass do BAC[y]:= BB [y];
      SmallBA:= CSmallBA;
      PoleBA:= CPoleBA;
      SawBA:= CSawBA;
    end;        {of SawPoleBA1}


  Procedure BugSawPoleBA;              {called from Grow}
    var y:integer;
    begin

SawPoleBA(BugClass,BugTPA,BugBA,BugSmallestClass,BugBiggestClass);
      for y:= BugSmallestClass to BugBiggestClass do BugBAC[y]:=
              BB[y];
      BugSmallBA:= CSmallBA;
      BugPoleBA:= CPoleBA;
      BugSawBA:= CSawBA;
    end;        {of SawPoleBA1}


  Procedure TSawPoleBA;
    var y:integer;
    begin
      SawPoleBA(ThinnedClass,ThinnedTPA,ThinnedBA,SmallestClass,Big
   gestClass);
      for y:= SmallestClass to BiggestClass do ThinnedBAC[y]:= BB [y];
      ThinnedSmallBA:= CSmallBA;
      ThinnedPoleBA:= CPoleBA;
      ThinnedSawBA:= CSawBA;
    end;         {of TSawPoleBA}


 Procedure BugTSawPoleBA;
    var y:integer;
    begin
      SawPoleBA(BugThinnedClass,BugThinnedTPA,BugThinnedBA,BugSmal
    lestClass,
      BugBiggestClass);
     for y:= BugSmallestClass to BugBiggestClass do BugThinnedBAC[y]:=
             BB [y];
      BugThinnedSmallBA:= CSmallBA;
      BugThinnedPoleBA:= CPoleBA;
      BugThinnedSawBA:= CSawBA;
    end;         {of BugTSawPoleBA}


   Procedure VolumeComputation (APoleBA:real; ASawBA:real;
       ASmallBA:real; ADomHt:real);    {called from Grow}

     {per acre volume estimates of the entire central stem, no top
                                                             limit}
    begin
       APoleCFV := 0.4085 * APoleBA * ADomHt;     {Buckman, 1962}
       ASawCFV := 0.4085 * ASawBA * ADomHt;
       ASmallCFV := 0.4085 * ASmallBA * ADomht;
       ATotalCFV := ASmallCFV + APoleCFV + ASawCFV;
```

```
{per acre volume estimates of the central stem to 3" top, dib}
    APoleCords := 0.003958 * APoleBA * ADomHt;   {Buckman, 1962}
    ASawCords := 0.003958 * ASawBA * ADomHt;
    AMerchCords := APoleCords + ASawCords;
    AMerchPoleCFV := 0.3127 * ADomHt * APoleBA;   {Lundgren, 1981}
    AMerchSawCFV := 0.3127 * ADomHt * ASawBA;
    AMerchCFV := AMerchPoleCFV + AMerchSawCFV;

    AMBFVolInt := (ASawCFV * BFperCF) / 1000;  {Lundgren, 1981}
    if AMBFVolInt < 0 then AMBFVolInt := 0;
    AMBFVolScrib := (2.084 * ASawBA * ADomHt) / 1000; {Buckman,
                                                          1962}
    if AMBFVolScrib < 0 then AMBFVolScrib := 0;
 end;        {of VolumeComputation}


Procedure VolComp1;
  begin
    VolumeComputation(PoleBA, SawBA, SmallBA, DomHt);
   PoleCFV := APoleCFV;  SawCFV := ASawCFV;  SmallCFV := ASmallCFV;

   TotalCFV := ATotalCFV;  PoleCords := APoleCords;  SawCords :=
              ASawCords;
   MerchCords := AMerchCords;  MerchPoleCFV := AMerchPoleCFV;
   MerchSawCFV := AMerchSawCFV;  MerchCFV := AMerchCFV;
   MBFVolInt := AMBFVolInt;  MBFVolScrib := AMBFVolScrib;
 end;   {end of VolComp1}


Procedure BugVolComp;
  begin
    VolumeComputation(BugPoleBA, BugSawBA, BugSmallBA, BugDomHt);
  BugPoleCFV := APoleCFV;  BugSawCFV := ASawCFV;  BugSmallCFV :=
              ASmallCFV;
  BugTotalCFV := ATotalCFV;  BugPoleCords := APoleCords;
  BugSawCords := ASawCords;
   BugMerchCords := AMerchCords;  BugMerchPoleCFV := AMerchPoleCFV;
   BugMerchSawCFV := AMerchSawCFV;  BugMerchCFV := AMerchCFV;
   BugMBFVolInt := AMBFVolInt;  BugMBFVolScrib := AMBFVolScrib;
 end;   {end of BugVolComp}


Procedure TVolComp;
  begin
    VolumeComputation(ThinnedPoleBA, ThinnedSawBA, ThinnedSmallBA,
                                                        DomHt);
   TPoleCFV := APoleCFV;  TSawCFV := ASawCFV;  TSmallCFV :=
              ASmallCFV;
   TTotalCFV := ATotalCFV;  TPoleCords := APoleCords;
   TSawCords := ASawCords;
   TMerchCords := AMerchCords;  TMerchPoleCFV := AMerchPoleCFV;
   TMerchSawCFV := AMerchSawCFV;  TMerchCFV := AMerchCFV;
   TMBFVolInt := AMBFVolInt;  TMBFVolScrib := AMBFVolScrib;
 end;   {end of TVolComp}


Procedure BugTVolComp;
  begin
   VolumeComputation(BugThinnedPoleBA, BugThinnedSawBA,
                              BugThinnedSmallBA, BugDomHt);
   BugTPoleCFV := APoleCFV;  BugTSawCFV := ASawCFV;
   BugTSmallCFV := ASmallCFV;
   BugTTotalCFV := ATotalCFV;  BugTPoleCords := APoleCords;
   BugTSawCords := ASawCords;
   BugTMerchCords := AMerchCords;  BugTMerchPoleCFV :=
                  AMerchPoleCFV;
   BugTMerchSawCFV := AMerchSawCFV;  BugTMerchCFV := AMerchCFV;
   BugTMBFVolInt := AMBFVolInt;  BugTMBFVolScrib := AMBFVolScrib;
 end;   {end of BugTVolComp}

Procedure BAG1;      {Lundgren, 1981;  stand age < 25 yrs.}
```

```
  begin
    x6 := exp(1.1677 * ln(1-exp(-0.040172 *(SeedAge+1))));
    x7 := 1-exp(-0.0018854 * TPA);
    x8 := exp(1.1677*ln(1-exp(-0.040172*SeedAge)));
    x9 := 1-exp(-0.0018854*TPA);
    BAGrowth1 := 6.5653*SI*(x6*x7-x8*x9);
  end;

Procedure BugBAG1;      {Lundgren, 1981;  stand age < 25 yrs.}
  begin
    x6 := exp(1.1677 * ln(1-exp(-0.040172 *(SeedAge+1))));
    x7 := 1-exp(-0.0018854 * BugTPA);
    x8 := exp(1.1677*ln(1-exp(-0.040172*SeedAge)));
    x9 := 1-exp(-0.0018854*BugTPA);
    BugBAGrowth1 := 6.5653*BugSI*(x6*x7-x8*x9);
  end;

Procedure BAG2;
  begin
    BAGrowth2 := 1.689 + (0.04107 * BA) - (0.000163 * sqr(BA)) -
                 (0.07696 * Seedage) + (0.0002274 * sqr(SeedAge)) +
                 (0.06441 * SI);
  end;

Procedure BugBAG2;
  begin
    BugBAGrowth2 := 1.689 + (0.04107 * BugBA) - (0.000163 *
          sqr(BugBA)) - (0.07696 * Seedage) + (0.0002274 *
          sqr(SeedAge)) + (0.06441 * BugSI);
  end;


Procedure ThinOption;           {called from ThinningOptions}
  begin
    clrscr;gotoxy(5,10);
    write('Do you want to thin this stand during this rotation  (Y
    or N)  ?  ');
    range := ['y','n','Y','N'];
    readln(ThinChoice);
    end;    {of ThinOption}


Procedure NM(ASmallestClass:integer; ABiggestClass:integer;
                                      AClass:p2; ADGR:real);
  var   n:real; Temp2,Temp1,x,y,z: integer;

  begin
  FillChar(AMortalityClass,SizeOf(AMortalityClass),0);
  FillChar(GAClass,SizeOf(GAClass),0);
  Temp1 :=0; Temp2 :=0; AMortalityBA := 0; AMortalityTPA :=0; CTPA
      :=0;
  if ASmallestClass <= 1 then z := 2 else Z := ASmallestClass;
  if ASmallestClass <= 1 then begin
    SurvivalRate := 0.9295;
    AMortalityClass[1] := AClass[1] - (round(AClass[1] *
                          SurvivalRate));
   CMortalityBA := CMortalityBA + (AMortalityClass[1] * (sqr(x) *
                   0.005454150));
    GAClass[1] := AClass[1] - AMortalityClass[1];
    temp1 := temp1 + GAClass[1];
    temp2 := temp2 + AMortalityClass[1];
  end;    {of if - then}
  for x := z to ABiggestClass do
    begin
      n:= 1.9953 + (57.97 * exp(1.012 * ln(ADGR))) + (0.2648 *
          exp(1.626 * ln(x-1))) * (exp(-0.1273 * (x-1)));
      SurvivalRate := 0.9997 - (1/(1+(exp(n))));
      AMortalityClass[x] := AClass[x] - (round(AClass[x] *
                            SurvivalRate));
      CMortalityBA := CMortalityBA + (AMortalityClass[x] * (sqr(x) *
                      0.00545415));
      GAClass[x] := AClass[x] - AMortalityClass[x];
```

```
      Temp1 := Temp1 + GAClass[x];
      Temp2 := Temp2 + AMortalityClass[x];
    end;    {of for - do loop}
  CTPA := Temp1;
  CMortalityTPA := Temp2;
  end;              {of NormalMortality}


Procedure NormalMortality;
  var y,z: integer;
  begin
    NM(SmallestClass,BiggestClass,Class,DGR);
    for y := SmallestClass to BiggestClass do MortalityClass[y] :=
      AMortalityClass[y];
    for z := SmallestClass to BiggestClass do Class[y] :=
          GAClass[y];
    TPA := CTPA;
    MortalityTPA := CMortalityTPA;
    MortalityBA := CMortalityBA;
    NMCFV :=0; NMSawCFV :=0; NMSawBF :=0;
  end;    {of NormalMortality}


Procedure BugNormalMortality;
  var y,z: integer;
  begin
    NM(BugSmallestClass,BugBiggestClass,BugClass,BugDGR);
    for y := BugSmallestClass to BugBiggestClass do
          BugMortalityClass[y] := AMortalityClass[y];
   for z := BugSmallestClass to BugBiggestClass do BugClass[y] :=
        GAClass[y];
    BugTPA := CTPA;
    BugMortalityTPA := CMortalityTPA;
    BugMortalityBA := CMortalityBA;
    BugNMCFV :=0; BugNMSawCFV :=0; BugNMSawBF :=0;
  end;    {of BugNormalMortality}


Procedure NMV(ASmallestClass:integer; ABiggestClass:integer;
 ASawClass:integer; AMortalityClass:p2); {Raile, et.al., 1982}

  Var    GSmallestClass,x,GSawClass : integer;
         temp1,temp2,temp3,temp4,a1,a2,a3,c1,c2,c3,d1,d2,d3 : real;

  begin
    temp1:=0; ANMCFV:=0; ANMSawcfv := 0; ANMSawBF:=0; temp2:=0;
    temp3:=0; temp4:=0; GSmallestClass:=0;
    a1:=114.3; c1:=0.07601; d1:=3.928;
    a2:=344.9; c2:=0.14728; d2:=7.536;
    a3:=63.09; c3:=0.12942; d3:=6.202;
    if ASmallestClass < 5 then GSmallestClass:=5 else
      GSmallestClass:= ASmallestClass;
    for x:= GSmallestClass to ABiggestClass do
      begin
      temp1:= exp(d1*ln(a1*(1-exp(-c1*x))));
      temp2:= AMortalityClass[x] * temp1;
      ANMcfv:= ANMcfv + temp2;    {4" top, outside bark}
      end;
  end;                {of NormalMortVol}


Procedure NormalMortVol;
  begin
    NMV(SmallestClass,BiggestClass,SawClass,MortalityClass);
    NMCFV := ANMCFV;
    NMSawCFV := ANMSawCFV;
    NMSawBF := ANMSawBF;
  end;

Procedure BugNormalMortVol;
  begin
```

```
NMV(BugSmallestClass,BugBiggestClass,SawClass,BugMortalityClass);
       BugNMCFV := ANMCFV;
       BugNMSawCFV := ANMSawCFV;
       BugNMSawBF := ANMSawBF;
     end;


   Procedure Thin;
     Var      frac: real; temp1: integer;

     begin
     frac:=0;
     case ThinningType of
     1: begin
        frac:= TreesToRemove/TPA;
        TPA:= TPA - TreesToRemove;
        ThinnedTPA:= TreesToRemove;
        ThinnedBA:= BA * frac;
        BA:= BA - ThinnedBA;
        end;     {of ThinningType 1}

     2: begin
        ThinnedBA:= BAProToRemove * BA;
        BA:= BA - ThinnedBA;
        ThinnedTPA:= round(BAProToRemove * TPA);
        TPA:= TPA - ThinnedTPA;
        end;      {of ThinningType 2}

     3: begin
        ThinnedBA:= BA - BAtoLeave;
        frac:= BAtoLeave/BA;
        BA:= BAtoLeave;
        Temp1:= round(frac * TPA);
        ThinnedTPA:= TPA - temp1;
        TPA:= temp1;
        end;      {of ThinningType 3}
     end;    {of case statement}
   end;   {of Thin}


   Procedure BugThin;
     Var      frac: real; temp1: integer;

     begin
     frac:=0; BugThinnedBA :=0; BugThinnedTPA:=0;
     case ThinningType of
     1: begin
        frac:= TreesToRemove/BugTPA;
        BugTPA:= BugTPA - TreesToRemove;
        BugThinnedTPA:= TreesToRemove;
        BugThinnedBA:= BugBA * frac;
        BugBA:= BugBA - BugThinnedBA;
        end;     {of ThinningType 1}

     2: begin
        BugThinnedBA:= BAProToRemove * BugBA;
        BugBA:= BugBA - BugThinnedBA;
        BugThinnedTPA:= round(BAProToRemove * BugTPA);
        BugTPA:= BugTPA - BugThinnedTPA;
        end;      {of ThinningType 2}

     3: begin
        BugThinnedBA:= BugBA - BAtoLeave;
        frac:= BAtoLeave/BugBA;
        BugBA:= BAtoLeave;
        Temp1:= round(frac * BugTPA);
        BugThinnedTPA:= BugTPA - temp1;
        BugTPA:= temp1;
        end;      {of ThinningType 3}
     end;    {of case statement}
   end;   {of Thin}
```

```
Procedure NST;

  begin
    NextScheduledThin:= StandAge + ThinningInterval;
  end;      {of NST}


Procedure BAGrowth;                  {called from Grow}
  begin
  BHAge;
    case SeedAge of
    1..24: begin
           BAG1;
           if BHA <= -1 then BA := BA + 0;
           if (BHA > -1) and (BHA <= 0) then
             BA := BA + (0.00545415 * TPA * (sqr(0.007 * SI *
                   (BHA+1)))));
           if BAGrowth1 < 0 then BA := BA + 0;
           if BAGrowth1 > MaxBAIncr then BA := BA + MaxBAIncr;
           if BAGrowth1 <= MaxBAIncr then BA := BA + BAGrowth1;
           end;
    25..250: begin
             BAG2;
             if BAGrowth2 < 0 then BA := BA + 0;
             if BAGrowth2 > MaxBAIncr then BA := BA + MaxBAIncr;
             if BAGrowth2 <= MaxBAIncr then BA := BA + BAGrowth2;
             end;
    end;     {of case}
  end;           {of BAGrowth}


Procedure BugBAGrowth;                 {called from Grow}
  begin
  BugBHAge;
    case SeedAge of
    1..24: begin
        BugBAG1;
         if BugBHA <= -1 then BugBA := BugBA + 0;
         if (BugBHA > -1) and (BugBHA <= 0) then
             BugBA := BugBA + (0.00545415 * BugTPA * (sqr(0.007 *
                      BugSI
              * (BugBHA+1)))));
         if BugBAGrowth1 < 0 then BugBA := BugBA + 0;
         if BugBAGrowth1 > BugMaxBAIncr then BugBA := BugBA +
               BugMaxBAIncr;
         if BugBAGrowth1 <= BugMaxBAIncr then BugBA := BugBA +
               BugBAGrowth1;
           end;
    25..120: begin
        BugBAG2;
         if BugBAGrowth2 < 0 then BugBA := BugBA + 0;
         if BugBAGrowth2 > BugMaxBAIncr then BugBA := BugBA +
               BugMaxBAIncr;
         if BugBAGrowth2 <= BugMaxBAIncr then BugBA := BugBA +
               BugBAGrowth2;
             end;
    end;     {of case}
  end;           {of BugBAGrowth}


Procedure menu1;           {called from procedure MainMenu}

  begin
  clrscr; gotoxy(5,5);
  writeln('Red pine growth & yield with no insects  . . . . . .
           . . . .  1');gotoxy(5,8);
  writeln('Red pine growth & yield with stochastic insect levels   .
           . . . .  2');gotoxy(5,11);
  writeln('Exit the program     . . . . . . . . . . . . .
           . . . .  3');
  repeat
```

```
  gotoxy(1,21);delline;gotoxy(10,21);
  write('Enter the number of your choice    ');
  range:= ['1'..'3'];
  readln(choice1);
  until Choice1 in [1..3];
end;     {of Menu1}


Procedure EditMenu1;                   {called from MainMenu}
  var   choice:char;

  begin
  repeat
    gotoxy(1,24); delline; gotoxy(10,24);
    write('Is your selection OK  (Y or N)?    ');
    range := ['n','y','N','Y'];
    readln(choice);
  case upcase(choice) of
    'N': Menu1;
  end;
  until upcase(choice) = #89;
  end;


Procedure InitialStandInfo;               {called from Basics}
  begin
  SetSI;
  SetPoleMin;
  SetSawMin;
  SetEstablishAge;
  SetAgeAtStart;
  SetTPA;
  SetBA;
  SetDisplayInterval;
  SetHarvestAge;
  end;                  {of InitialStandInfo}


Procedure SP;                  {called from BugBasics}
  begin
    SetSitePrep;
    SetSitePrepCost;
    SetPlantingMethod;
    SetPlantingMethodCost;
    SetWoodPrice;
    SetInventoryCost;
    SetLayoutCost;
    SetInterestRate;
  end;   {of SP}


Procedure Basics;                      {called from Main1}
  begin
    FillChar(r,SizeOf(r),0);
    FillChar(t,SizeOf(t),0);
    InitialStandInfo;
    EditISI;
  end;      {of Basics}


Procedure BugBasics;                      {called from Main2}
  begin
    FillChar(r,SizeOf(r),0);
    FillChar(t,SizeOf(t),0);
    FillChar(bugr,SizeOf(bugr),0);
    FillChar(bugt,SizeOf(bugt),0);
    FillChar(econ,SizeOf(econ),0);
    ChooseBug;
    InitialStandInfo;
    EditISI;
    SP;
```

```
    EditSitePrep;
    SetSeedlingSurvival;
  end;    {of BugBasics}


Procedure ThinningOptions;
  begin
    ThinOption;
    if upcase(ThinChoice) = #89 then
    begin
    ThinningTiming;
    AThinningType;
    EditThinningOptions;
    end;
    end;     {of ThinningOptions}


Procedure StoreValues; forward;

Procedure RunOptions; forward;

Procedure MainMenu; forward;

Procedure StoreThinValues; forward;

Procedure StoreBugValues; forward;

Procedure StoreBugThinValues; forward;

Procedure CountMortTrees; forward;


Procedure GrowIt;
    begin
      QMD1;
      MDI1;
      DGRate;
      MBAI1;
      DH1;
      DSD1;
      BAGrowth;
      NumbersPerClass1;
      NormalMortality;
      SawPoleBA1;
      ConvertBFtoCF1;
      VolComp1;
      if MortalityTPA > 0 then NormalMortVol;
      StoreValues;
    end;   {of GrowIt}


Procedure BugGrowIt;
    begin
      BugQMD;
      BugMDI;
      BugDGRate;
      BugMBAI;
      BugDH;
      BugDSD;
      BugBAGrowth;
      BugNumbersPerClass;
      BugNormalMortality;
      BugSawPoleBA;
      BugConvertBFtoCF;
      BugVolComp;
      if BugMortalityTPA > 0 then NormalMortVol;
      StoreBugValues;
    end;   {of BugGrowIt}


Procedure ThinIt;
  begin
```

```
    Thin;
    TNumbersPerClass;
    TSawPoleBA;
    TVolComp;
    StoreThinValues;
  end;   { of ThinIt }


Procedure BugThinIt;
  begin
    BugThin;
    BugTNumbersPerClass;
    BugTSawPoleBA;
    BugTVolComp;
    StoreBugThinValues;
  end;   { of BugThinIt }


Procedure OtherThin;
  begin
    Display1;
    ThinningYearDisplay;
    WaitaBit;
    EditThinningOptions;
    RunOptions;
    NST;
   end;  {of OtherThin}


Procedure BugOtherThin;
  begin
    BugDisplay1;
    BugThinningYearDisplay;
    WaitaBit;
    EditThinningOptions;
    RunOptions;
    NST;
   end;  {of BugOtherThin}


Procedure Grow;            {called from Main1}
 var  y,count,x: integer;
  begin
  NextScheduledThin := MinTAge - 1;
  x := 1;
  clrscr;
      count := 0;
      repeat
      BHAge;
      if BHA1 > -1 then begin
  if (BA = 0) and (SeedAge >=4) and
   (SeedAge < 25) then begin
   BAU1;
   BA :=BAUnder25;
   end;
  repeat
    y := 0;
    if r[StandAge - 1].BA = 0 then begin
    y:= count - (displayinterval * (trunc(count/displayinterval)));
    x := y + 1;
    end
    else
    x := 1;
 while (x <= DisplayInterval) and (StandAge < HarvestAge) do begin
      clrscr; gotoxy(19,12);
      writeln('Currently doing calculations for year  ',StandAge);
      gotoxy(79,25);
        GrowIt;
      if StandAge = NextScheduledThin then
        begin
        ThinIt;
        OtherThin;
```

```
        end;    {of thinning if - then}
      StandAge := StandAge + 1;
      SeedAge := StandAge + AgeAtEst;
      x := x + 1;
    end;      {of while - do}
      if ((StandAge - 1) <> (NextScheduledThin - ThinningInterval))
        or
      ((StandAge - 1) < MinTAge) or (StandAge > HarvestAge) then
        begin
      if (BHA1 > -1) and (r[StandAge-1].BA <> 0) then begin
        CountMortTrees;
        Display1;
        CurrentDisplay;
        WaitABit;
        RunOptions;
        end;   { of if BHA1 > -1 then}
      end;  {of if - then}
  until StandAge >= HarvestAge;
  end;   {of major if BHA1 > -1 then}
  StandAge := StandAge + 1;
  SeedAge := StandAge + AgeAtEst;
  count := count + 1;
  until StandAge >= HarvestAge;
  end;      {of Grow}


Procedure BugGrow;              {called from Main2}
 var  y,count,x: integer;
  begin
  BHA1 :=0; BugBHA1 :=0;
  NextScheduledThin := MinTAge - 1;
  StartTPA1 := TPA;   StartTPA2 := BugTPA;
  x := 1;
  clrscr;
      count := 0;
      repeat
      BHAge;  BugBHAge;
      if (BHA1 > -1) or (BugBHA1 > -1) then begin
  if (BA = 0) and (SeedAge >=4) and
   (SeedAge < 25) then begin
    BAU1; BugBAU;
    BA :=BAUnder25;
    BugBA := BugBAUnder25;
    end;
  repeat
    y := 0;
    if r[StandAge - 1].BA = 0 then begin
    y:= count - (displayinterval * (trunc(count/displayinterval)));
    x := y + 1;
    end
    else
    x := 1;
 while (x <= DisplayInterval) and (StandAge < HarvestAge) do begin
      clrscr; gotoxy(19,12);
      writeln('Currently doing calculations for year  ',StandAge);
      gotoxy(79,25);
        GrowIt;
      if StandAge = NextScheduledThin then
        begin
        ThinIt;
        if BugBHA1 <= -1 then OtherThin;
        end;    {of thinning if - then}
      if (BugBHA1 > -1) and (StandAge < BugHarvestAge) then begin
        BugGrowIt;
      if StandAge = NextScheduledThin then begin
        BugThinIt;
        BugOtherThin;
        econ[StandAge].inventory := InventoryCost;
        econ[StandAge].layout := LayoutCost;
      end;    {of bug thinning if - then}
      end;   {of if BugBHA1 > -1 if - then}
      StandAge := StandAge + 1;
```

```
        SeedAge := StandAge + AgeAtEst;
        x := x + 1;
      end;     {of while - do}
        if ((StandAge - 1) <> (NextScheduledThin - ThinningInterval))
             or
        ((StandAge - 1) < MinTAge) or (StandAge > HarvestAge) then
             begin
        if (BHA1 > -1) and (r[StandAge-1].BA <> 0) then begin
          CountMortTrees;
          BugDisplay1;
          BugCurrentDisplay;
          WaitABit;
          RunOptions;
          end;   { of if BHA1 > -1 then}
        end;  {of if - then}
    until StandAge >= HarvestAge;
    end;   {of major if BHA1 > -1 then}
    StandAge := StandAge + 1;
    SeedAge := StandAge + AgeAtEst;
    count := count + 1;
    until StandAge >= HarvestAge;
    end;      {of BugGrow}


  Procedure StoreValues;      {called from Grow}
    var   x: integer;

      begin
      x:= StandAge;
      r[x].StandAge := StandAge;
      r[x].TPA := TPA;
      r[x].BA := BA;
      r[x].AveBA := AveBA;
      r[x].QMdbh := QMdbh;
      r[x].DGR := DGR;
      r[x].DomHt := DomHt;
      r[x].PoleCFV := PoleCFV;
      r[x].SawCFV := SawCFV;
      r[x].SmallCFV := SmallCFV;
      r[x].TotalCFV := TotalCFV;
      r[x].PoleCords := PoleCords;
      r[x].SawCords := SawCords;
      r[x].MerchCords := MerchCords;
      r[x].MerchPoleCFV := MerchPoleCFV;
      r[x].MerchSawCfv := MerchSawCFV;
      r[x].MerchCFV := MerchCFV;
      r[x].MBFVolInt := MBFVolInt;
      r[x].MbfVolScrib := MBFVolScrib;
      r[x].MortalityTPA := MortalityTPA;
      r[x].NMcfv := NMcfv;
    end;        {of StoreValues}


  Procedure StoreBugValues;      {called from Grow}
    var   x: integer;

      begin
      x:= StandAge;
      bugr[x].StandAge := StandAge;
      bugr[x].TPA := bugTPA;
      bugr[x].BA := bugBA;
      bugr[x].AveBA := bugAveBA;
      bugr[x].QMdbh := bugQMdbh;
      bugr[x].DGR := bugDGR;
      bugr[x].DomHt := bugDomHt;
      bugr[x].PoleCFV := bugPoleCFV;
      bugr[x].SawCFV := bugSawCFV;
      bugr[x].SmallCFV := bugSmallCFV;
      bugr[x].TotalCFV := bugTotalCFV;
      bugr[x].PoleCords := bugPoleCords;
      bugr[x].SawCords := bugSawCords;
      bugr[x].MerchCords := bugMerchCords;
```

```
    bugr[x].MerchPoleCFV := bugMerchPoleCFV;
    bugr[x].MerchSawCfv := bugMerchSawCFV;
    bugr[x].MerchCFV := bugMerchCFV;
    bugr[x].MBFVolInt := bugMBFVolInt;
    bugr[x].MbfVolScrib := bugMBFVolScrib;
    bugr[x].MortalityTPA := bugMortalityTPA;
    bugr[x].NMcfv := bugNMcfv;
  end;          {of StoreValues}


  Procedure StoreThinValues;          {called from Grow}
  var   x: integer;

    begin
    x:= StandAge;
    t[x].StandAge := StandAge;
    t[x].ThinnedTPA := ThinnedTPA;
    t[x].ThinnedBA := ThinnedBA;
    t[x].TPoleCFV := TPoleCFV;
    t[x].TSawCFV := TSawCFV;
    t[x].TSmallCFV := TSmallCFV;
    t[x].TTotalCFV := TTotalCFV;
    t[x].TPoleCords := TPoleCords;
    t[x].TSawCords := TSawCords;
    t[x].TMerchCords := TMerchCords;
    t[x].TMerchPoleCFV := TMerchPoleCFV;
    t[x].TMerchSawCfv := TMerchSawCFV;
    t[x].TMerchCFV := TMerchCFV;
    t[x].TMBFVolInt := TMBFVolInt;
    t[x].TMbfVolScrib := TMBFVolScrib;
  end;          {of StoreThinValues}


 Procedure StoreBugThinValues;          {called from Grow}
  var   x: integer;

    begin
    x:= StandAge;
    bugt[x].StandAge := StandAge;
    bugt[x].ThinnedTPA := bugThinnedTPA;
    bugt[x].ThinnedBA := bugThinnedBA;
    bugt[x].TPoleCFV := bugTPoleCFV;
    bugt[x].TSawCFV := bugTSawCFV;
    bugt[x].TSmallCFV := bugTSmallCFV;
    bugt[x].TTotalCFV := bugTTotalCFV;
    bugt[x].TPoleCords := bugTPoleCords;
    bugt[x].TSawCords := bugTSawCords;
    bugt[x].TMerchCords := bugTMerchCords;
    bugt[x].TMerchPoleCFV := bugTMerchPoleCFV;
    bugt[x].TMerchSawCfv := bugTMerchSawCFV;
    bugt[x].TMerchCFV := bugTMerchCFV;
    bugt[x].TMBFVolInt := bugTMBFVolInt;
    bugt[x].TMbfVolScrib := bugTMBFVolScrib;
  end;          {of StoreBugThinValues}


Procedure RunOptions;                    {called from Grow}
  begin
  clrscr;
  gotoxy(5,6); write('1.  CONTINUE with this projection.');
  gotoxy(5,8); write('2.  CHANGE harvest age.');
  gotoxy(5,10); write('3.  START a thinning treatment.');
  gotoxy(5,12); write ('4.  REPEAT the previous output screen.');
  gotoxy(5,14); write('5.  CHANGE the display interval.');
  gotoxy(5,16); write('6.  QUIT & Return to Main Menu at beginning
                       of RPPEST.');
  gotoxy(10,20); write('Enter your choice of these options.   ');
  range :=['1'..'6'];
  readln(ROChoice);
    case ROChoice of
    2: begin
        if BugChoice <> 3 then begin
```

```
        SetHarvestAge;
        RunOptions;
        end;  { of if - then }
      end;
    3: begin
        if (upcase(ThinChoice) = #78) and (StandAge < HarvestAge)
            and
        (upcase(ThinChoice) = 'N') then begin
        ThinningTiming;
        AThinningType;
        EditThinningOptions;
        NextScheduledThin := MinTAge;
        end;  { of if - then}
        RunOptions;
      end;
    4: begin
      if Choice1 = 1 then begin
        Display1;
        CurrentDisplay;
        Waitabit;
        RunOptions;
        end;   { of Choice 1 }
        if Choice1 = 2 then begin
        BugDisplay1;
        BugCurrentDisplay;
        Waitabit;
        RunOptions;
        end;    { of Choice 2 }
      end;
    5: begin
      SetDisplayInterval;
      RunOptions;
      end;
    6: StandAge := HarvestAge + 1;
    end;  {of case}
  end;   {of RunOptions}


Procedure ZeroOut;      {called from Main1 & Main2}
   begin
     SI :=0;  PoleMin :=0;  SawMin :=0;  AgeAtEst :=0;
     StartAge :=0;  BA :=0;  TPA :=0;  HarvestAge :=0;
     DisplayInterval :=0;
     ThinningType :=0;  TreesToRemove :=0;  BAProToRemove :=0;
     MaxTAge :=0;
     BAToLeave :=0;  MinTAge :=0;  MinTBA :=0;  ThinningInterval
     :=0;
     NextScheduledThin :=0; BugSI :=0; BugBA :=0; BugTPA :=0;
     BugHarvestAge := 0;
   end;


Procedure CompInt;
  begin
    DiscountedValue := 0;
    DiscountedValue := value / (EXP((age + 1) * ln( 1 +
                       InterestRate)));
  end;   { of CompInt }

Procedure CountMortTrees;
  var    x: integer;
  begin
  MortalityTreeCount:=0; BugMortalityTreeCount:=0; MortVolSum :=0;
  BugMortVolSum :=0;
  for x := 0 to StandAge do begin
    with r[x] do begin
      if MortalityTPA > 0 then begin
      MortalityTreeCount := MortalityTreeCount + MortalityTPA;
      MortVolSum := MortVolSum + NMCFV;
      end;  { of if - then }
    end;    { of with - do }
    with bugr[x] do begin
```

```
      if MortalityTPA > 0 then begin
      BugMortalityTreeCount := BugMortalityTreeCount + MortalityTPA;
      BugMortVolSum := BugMortVolSum + NMCFV;
      end;  { of if - then }
    end;    { of with - do }
  end;    { of for - do }
end;   { of CountMortTrees }


Procedure ImplementBug;
  begin
  case BugChoice of
    1: Grubs;
    2: Weevil;
    3: Spittlebug;
    4: Sawfly;
  end;  {of case stmt.}
end;   {of ImplementBug}


Procedure Main1;                      {called from MainMenu}

  begin
    ZeroOut;
    Basics;
    ThinningOptions;
    Grow;
    if ROChoice <> 6 then begin
    VolumeSummary;
    DisplayVolumeSummary;
    WaitABit;
    end;  { of if then }
  end;  { of Main1 }

Procedure Main2;                      {called from MainMenu}
  begin
    ZeroOut;
    BugBasics;
    ImplementBug;
    ThinningOptions;
    BugGrow;
    if ROChoice <> 6 then begin
    NPV;
    VolumeSummary;
    DisplayVolumeSummary;
    WaitABit;
    DisplayBugVolumeSummary;
    WaitABit;
    DisplayEcon;
    WaitABit;
    end;  { of if then }
  end;  { of Main 2 }


Procedure MainMenu;                   {called from main program}

  begin
    repeat
    Menu1;
    EditMenu1;
    case choice1 of
      1: Main1;
      2: Main2;
      3: begin
         clrscr; gotoxy(17,10);
         writeln('Normal termination of Program RPPest.');
         end;
    end;
    until choice1 = 3;
    end;
```

```
BEGIN                        {  **  Main Program  **  }
storeptr := Coninptr;
coninptr := ofs(ConIn);
autoupcase := true;
MainMenu;
end.
```