THESIS

This is to certify that the

thesis entitled

A simulation of jack pine budworm and white pine weevil monitor and control techniques and their effects on jack pine economics.

presented by

Raymond James Drapek

has been accepted towards fulfillment
of the requirements for

Masters degree in Forestry

Gary A. Simmon
Major professor

Gary A. Simmon

Date Feb. 11, 1985

A SIMULATION OF JACK PINE BUDWORM AND WHITE PINE WEEVIL MONITOR AND CONTROL TECHNIQUES AND THEIR EFFECTS ON JACK PINE ECONOMICS

By

Raymond James Drapek

A THESIS

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

MASTER OF SCIENCE

Department of Forestry

1985

ABSTRACT

A SIMULATION OF JACK PINE BUDWORM AND WHITE PINE WEEVIL MONITOR AND CONTROL TECHNIQUES AND THEIR EFFECTS ON JACK PINE ECONOMICS

By

Raymond James Drapek

A model was developed to simulate volume growth of jack pine (Pinus banksiana Lamb.) under different intensities of attack by the jack pine budworm (Choristoneura pinus Freeman) and the white pine weevil (Pissodes strobi Peck.). The model allows a user to determine the site index, number of acres, discount rate, the deterministic or stochastic nature of the run, management costs, level of weevil damage, and the methods of site preparation and regeneration. The user also can control what kind of monitor and spray policy to use for the insects and when to cut the stand. Final results of the program would include the amount of wood harvested and the amount saved by insect control efforts. These would be compared with discounted costs to obtain a net present value for the stand and a net present value for insect control efforts at stand initiation.

To my grandfather,

Walter Daniel Heaphy

ACKNOWLEDGMENTS

A sincere thanks to all who helped me complete this project:

To my advisor, Dr. Gary Simmons, for easy-going support and valuable advice;

To Frank Sapio and Ron Murray, for making their considerable knowledge of forest pest management available;

To Dr. Stuart Gage, Aubrey Moore, Bob Kriegel, and Jim Pieronek, for much-needed assistance on the computer;

To Dr. Robert Marty and Dr. Donald Dickmann, for useful feedback on the project as a whole;

To Norm Elliott, for putting up with all my complaining;

To Susan Battenfield, for making me for to rite real good;

To Phil Deschaine and Claudia Haas, for sharing with me their friendship and some tasty quesadillas;

And finally, to my parents, Joseph and Doris Drapek, for never doubting that I could do it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Jack pine (Pinus banksiana Lamb.) is a major timber species of the Lake States Region. Short-lived and shade-intolerant, it is usually found growing on dry sites that have been burned over. Though it often grows alone, it can be found in association with the following (Fowells, 1965):

red pine (Pinus resinosa, Ait)

eastern white pine (Pinus strobus L.)

quaking aspen (Populus tremuloides Mich.)

big-tooth aspen (Populus grandidentata Mich.)

paper birch (Betula papyrifera Marsh)

red oak (Quercus rubra L.)

northern pin oak (Quercus ellipsoidalis E.J. Hill)

black spruce (Picea mariana (Mill) B.S.P.)

white spruce (Picea glauca (Moench) Vass)

On better sites, jack pine is transitional and is soon replaced by shade-tolerant species. Once considered worthless, it now is valued for its uses as pulp, sawtimber, and as wildlife habitat. As of 1980, over 35 million dollars worth of jack pine was growing on 836,700 acres in the Lake States.

Two problems which arise in managing for jack pine are (1) getting adequate regeneration and (2) protecting the trees against insect pests. Site conditions required for optimal establishment and growth include bare mineral soil for adequate seed germination and minimal competition from weeds or shrubs. This means that a site usually must undergo some preparation before jack pine will regenerate. Management activities designed to produce these optimal conditions include full tree skidding, hand cutting of competing

vegetation, hand or machine scalping, rollerchopping, bulldozing, raking, prescribed burning, and herbicides (Benzie, 1977; Hacker et al., 1983).

Two insects of concern in jack pine management are the jack pine budworm (Choristoneura pinus Freeman) and the white pine weevil (Pissodes strobi Peck.). Larvae of the jack pine budworm feed on the buds and the foliage of jack pine trees. When budworm populations are high this feeding can cause mortality and growth loss to merchantable trees and seedlings. As a result, jack pine stands can become understocked and slow growing (Dixon, 1981). The white pine weevil attacks jack pine leaders. When the leader is killed, the tree grows bushy and crooked. Whole stands can be made unmerchantable as a result of such damage.

Technically, it is possible to always assure adequate regeneration and to protect trees against insect attack. Mechanical and chemical site preparation techniques are effective when correctly applied, and insecticides provide adequate short-term control. The problem in jack pine management is that the margin of profit is small even under the best of conditions, and the costs of active management can easily push the entire project into the red. Therefore, a management strategy must balance the loss that will occur if nothing is done and the loss that will occur if too much is done.

The purpose of the model outlined in this thesis is to evaluate the economic efficacy of jack pine insect control programs. In the model, the growth of a jack pine stand is simulated. Budworm and weevil populations and their effects on stand growth are also simulated. These populations are modified by insect control efforts that can be set for each run of the model. The final result of the

model includes an evaluation of the net present worth of the stand and the net present worth of control efforts at the time of stand initiation.

A second purpose of this project was to develop a computer program of the model to be used as an educational tool by people who are involved in managing forests for timber. The program will allow them to monitor and control insects on a hypothetical stand and have their efforts evaluated economically. Hopefully, the difficulties of balancing control related costs against the cost of not controlling will be apparent after a few simulated rotations.

THE MODEL

An Overview

The model can be divided into two basic phases. The first phase represented by the left half of figure 1 covers the initializing of model parameters and the first 20 years growth of the stand. The second phase represented by the right half covers stand growth from year 20 to the final harvest. This division was necessary because growth equations were only available for stands age 20 and over. This meant that the results of all events occurring in the first 20 years would somehow have to be accumulated and used at year 20 to produce an initial stand. Stand growth afterwards could be modeled yearly.

In both phases of the model, potential insect levels are set for every year. Insect populations can then sampled if monitoring activity is chosen for that year. If insect control is chosen, the severity of insect attack can be modified. Tree damage, is then determined based on insect attack severity. Before year 20,

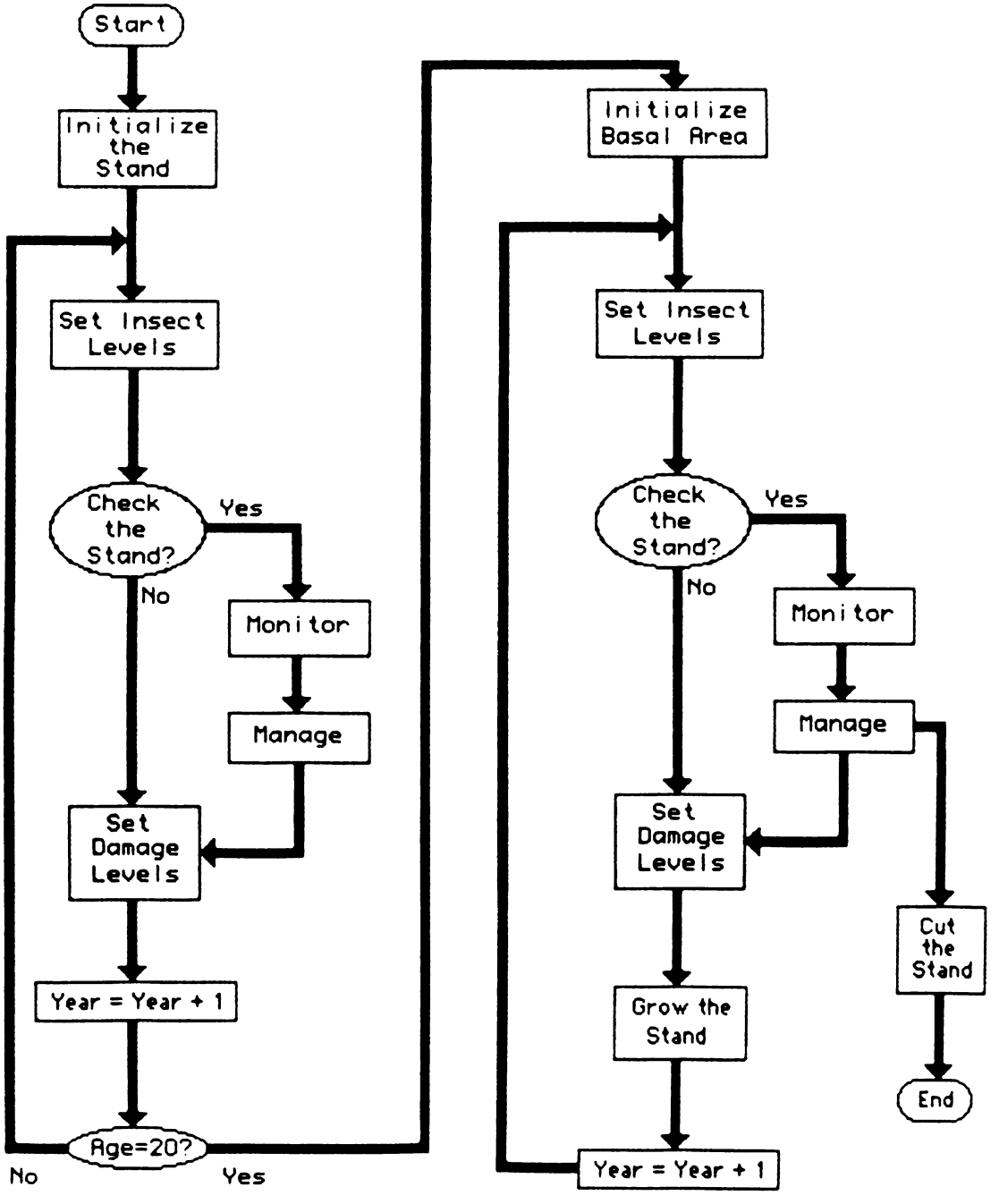


Figure 1. A Flowchart of the Model.

insect-caused tree damage is accumulated and used to modify the initial setting of stand basal area. After year 20, damage is applied yearly. When the stand is cut, an economic evaluation of all previous activities occurs and the model is exited.

Initialization

During initialization, parameters that will affect management costs, tree growth, and insect populations are set. Figure 2 lists everything that can be set. Among the options is one to make the run automatic or interactive. In automatic run, the user sets the rules for monitoring and control activities in the beginning, and the computer automatically carries these out during the run.

Setting Insect Levels

Setting Budworm Levels. Jack pine budworm populations cycle between outbreak levels where tree damage can occur and endemic levels where populations are often undetectable. Outbreaks last approximately four years and are characterized by a steady rise and then a sharp drop. Cycles vary in length from 6 to 15 years. In this model, a generic number ranging from zero to ten is used to represent the population levels of the budworm. A zero signifies an endemic population and a ten represents the maximum population possible. This generic number is converted in the sampling portion of the model to actual insect numbers per unit sample.

This cyclic pattern is simulated in the model by (1) randomly determining the number of years between outbreaks and then (2) randomly generating population levels for each of four outbreak years. The number of years between

- Number of Acres in the stand

- Site Index

- Weevil levels to expect

- Costs: 1) Costs of Site Preparation Activities
  2) Costs to Plant or Seed the Stand
  3) Costs of insecticide applications
  4) Wages of Forest Technicians and Pest Scouts

- Type of Site Preparation to Use

- Method of Regeneration to use

- Whether Model Run will be Random or Deterministic

- Whether Model will Run Automatically or Interactively

- If Automatic: Set up a Monitor/Manage Plan for Budworm and Weevil
  1) Decide if will Monitor and/or Spray
  2) Decide when will Monitor and/or Spray
  3) Decide what and how will Spray

- If Automatic: Determine Rotation Length

Figure 2. Items Which Can Be Varied During The Initialization Phase.

outbreaks was established using a truncated exponential distribution with a mean of six years and truncation points of three and eleven years. A random variable with this distribution can be generated as follows:

$$X=-U(\ln r)$$

where U is the mean (6) and r is a uniformly distributed random variable between 0 and 1 (Manetsch and Park, 1980). This method of simulating the timing of jack pine budworm outbreaks was also used by Nyrop et al. (1983). This model diverges from their model as to how it simulates the effects of budworm defoliation on stand growth.

A population level is determined for each of the four outbreak years by setting upper and lower population levels and generating a uniformly distributed number between those two values. Upper and lower values were chosen for each year so that populations would rise and fall as a natural population would during an outbreak. Levels were set so that very low to moderate populations would occur during the first year, moderate to high populations would occur during the second year, moderate to very high populations would occur during the third year, and very low to very high populations would occur during the last year of the outbreak. Generic population values correspond to population level descriptions in the following manner: levels of 0 to 3 correspond to very low populations, levels of 4 and 5 correspond to low populations, levels of 6 and 7 correspond to moderate populations, a level of 8 corresponds to high populations, and a 9 to very high populations. For those years between outbreaks, a population value of zero is given.

Setting Weevil Levels. Unlike the jack pine budworm, white pine weevil populations do not display distinct cycles. Weevil populations are fairly steady within any one stand, but most observations to date suggest that their populations tend to increase with time (Marty and Mott, 1964). Weevil populations can be enumerated by noting what percent of pine leaders have been attacked. Observed values of weevil attack have ranged from 0 to close to 100 percent. In this model, the level of weevil attack is set in the initialization phase. Weevil levels grow linearly to this level for the first ten years and remain there for another ten years. After year 20, the weevil is no longer considered a problem in jack pine and weevil populations are no longer modeled.

Monitoring Insect Populations

Many alternatives currently are available for sampling the jack pine budworm (Dixon, 1980). Two systems are currently used for estimating egg, early larval, late larval, and pupal populations: the 15-inch branch sample (Minnesota Department of Natural Resources) and the cluster system (Foltz et al., 1968). Overwintering larval surveys and pheromone traps for adults are also routinely used. To simplify coding the model for computer use, the number of sampling methods simulated was limited. An overwintering larval sample and a late larval sample were chosen to provide an early warning of potential trouble and a final pre-damage stand check.

In the overwintering larval survey currently used at Michigan State University, 18-inch branches are taken from the stand during the winter months when larvae are in their hibernaculum. The branches are sent to a lab where they are cut up and soaked in a 0.05% solution of sodium hydroxide. This process

separates out the larvae, which are then collected on filter paper and counted. This sampling only serves as a crude early warning of potential trouble. Finding any larvae on a branch indicates that populations may become high in that particular stand.

Two problems came up in attempting to simulate this sampling method: (1) no precise measures are available on the precision or accuracy of this sampling method and (2) no sure way is available to relate population levels at the time of sampling with population levels during the damaging stages of the insect's life cycle. Since little correlation has occurred in the past between this sampling and later sample results, a maximum error for the difference between the results of this sampling and the final population of the budworm during damaging stages was chosen at 70 percent. This error is applied to the actual population level by randomly generating a number and adding or subtracting that number from the true population level. A uniform random distribution, maximum error of 70 percent, is used. This result is then converted to appropriate sample values by assigning values of zero larvae per 18-inch branch for all results less than four, seven larvae per 18-inch branch for all results greater than nine, and linearly assigning values for all results between four and nine.

The late larval survey used is the same as that presented by Dixon (1980) for 15-inch branch samples. For each stand two, 15-inch branch samples are taken from three trees and are inspected for 5th and 6th instar budworm larvae. An average of three or more larvae per branch indicates a high population. For simulating this sampling method a 10 percent error was chosen as a reasonable representation of actual sampling errors. Sample values are determined in a similar manner to the overwintering larval survey described above. For error-

adjusted populations of less than zero, a sample value of zero larvae per 15-inch branch is assigned. Error-adjusted populations greater than eight result in a sample value of three larvae per 15-inch branch. Once again, interim values are assigned linearly.

White pine weevil damage is sampled for by observing the percentage of leaders that have been killed. Since weevil levels also are measured within the model on a percent leaders-hit basis, no transformation is required to go from model-generated weevil levels to model-generated weevil samples. A 10 percent error, however, is included in the model to simulate sampling error.

Control of Insect Populations

Control options for the jack pine budworm and the white pine weevil within the model are limited to insecticides. For both insects, there is the option to apply the pesticide aerially or to go with a ground application. For the jack pine budworm an additional option is available to go with a biological pesticide, such as Bacillus thuringiensis, or to go with a chemical pesticide, such as Carbaryl. Costs of application and efficacies of population control differ between the alternatives. As of 1984, no solid information was available on the efficacies of the various alternatives. Best-guess estimates of the ranges of insect kills obtained from each method are used in the model. These are listed in Table 1. When an insecticide is applied in the model, a value is randomly chosen from within this range. Though default spray costs are on a simple per acre basis, spray costs can be set up as an initial up-front cost to initiate the operation plus an additional per acre cost. This allows economies of scale to operate within the model. It then is cheaper per acre to spray a large stand than it is to spray a

TABLE 1. Percent reductions in white pine weevil and jack pine budworm levels as a result of various spray operations.

Type of Spray Operation	Population reduction. Year of Spray.	Population Reduction Year After Spray
Aerial application of a chemical pesticide directed against budworm	90 - 100%	0%
Aerial application of a biological pesticide directed against budworm	70 - 90%	0%
Ground application of a chemical pesticide directed against budworm	90 - 100%	0%
Ground application of a biological pesticide directed against budworm.	70 - 90%	0%
Aerial application of a chemical pesticide directed against weevil.	40 - 50%	15 - 35%
Ground application of a chemical pesticide directed against weevil	40 - 50%	15 - 35%

TABLE 2. Costs of various insecticide applications.

Type of Spray Operation	Initial Cost	Cost Per Acre
Aerial application of a chemical pesticide directed against budworm	$ 0.00	$ 8.00
Aerial application of a biological pesticide directed against budworm	$ 0.00	$ 10.00
Ground application of a chemical pesticide directed against budworm	$ 0.00	$ 20.00
Ground application of a biological pesticide directed against budworm	$ 0.00	$ 25.00
Aerial application of a chemical pesticide directed against weevil	$ 0.00	$ 8.00
Ground application of a chemical pesticide directed against weevil	$ 0.00	$ 30.00

small one. The values used in the model for these various costs are listed in Table 2.

Setting Insect Caused Tree Damage

After all control efforts have been completed and final insect population levels are set, the amount of insect caused tree damage is set. There are three types of tree damage: growth loss, mortality, and loss of merchantability. Growth loss and mortality are related to jack pine budworm population levels. Physiological stress caused by budworm feeding damage reduces growth and increases the likelihood of death. The loss of merchantability is related to weevil population levels. The death of the leader causes bushy and/or crooked growth for the tree making its removal from the stand more difficult. If the tree is hit too many times, its value can be reduced to nothing.

Growth Loss. Growth losses were estimated using values derived from Table 3 (Kulman et al., 1963). By assuming that the ratio of latewood to earlywood in jack pine averages around 1:2.5 (Kulman et al., 1963), the values from Table 4 were generated. Table 4 contains not only the proportion of normal growth that will occur from the current years' defoliation, but also the proportions of normal growth resulting from defoliation over the previous two years. Growth loss appears to extend beyond the three years covered here for some defoliation classes, but no attempt was made to model this extension since no information was available on losses three or more years after defoliation. The defoliation classes used in Tables 3 and 4 were based on visual inspections of trees. Kulman et al. (1963) crudely related these classifications to results of a

TABLE 3. Percent reduction in average growth ring area in jack pine resulting from budworm defoliation in 1956. (Taken from Kulman et al., 1963)

Defoliation class	Early wood		Late wood		
	1957	1958	1956	1957	1958
2	54	8	32	27	0
3	76	1	60	44	0
4	100	52	83	73	20
5	100	86	106	91	86

TABLE 4. Proportion of normal growth resulting from jack pine budworm defoliation.

Defoliation class	1st year	2nd year	3rd year
1	1.00	1.00	1.00
2	0.92	0.53	0.94
3	0.84	0.33	0.99
4	0.78	0.07	0.54
5	0.73	0.02	0.14

Defoliation class 1 = Very Light
Defoliation class 2 = Light
Defoliation class 3 = Moderate
Defoliation class 4 = High
Defoliation class 5 = Very High

tree-beating sampling method, but no information was available on how to relate their sampling method with those currently used. Levels were assigned in the following manner for this model: 1 = very low, 2 = low, 3 = moderate, 4 = high, and 5 = very high. Table 4 is used in the model by the following procedure:

1. For every year of outbreak choose one of the five defoliation classes based on that year's population level. (populations of 0-3 = very light, 4-5 = light, 6-7 = medium, 8 = heavy, 9 = very heavy).

2. For each of the three years obtain a value from Table 4 by cross-referencing the defoliation class with the year under consideration.

3. Multiply these three values together to obtain the proportion of normal growth that will occur that year.

For example, if a heavy defoliation occurs this year and was preceeded by a light defoliation last year and a very light defoliation the year before, then 41 percent of normal growth will occur this year. Values from Table 4 are used in the following manner to produce this result:

$$(0.78 \times 0.53 \times 1.00 = 0.41)$$

Yearly growth loss for stands over 20years old is simulated by multiplying the volume growth obtained from the tree growth portion of the model with the growth proportion obtained above. For stands under 20 years old this cannot be done since stand growth is not simulated yearly. Therefore, the proportion of normal stand growth is averaged over the first 20 years and then is applied when the initial basal area is set at year 20. Otherwise, growth loss is treated identically over all ages.

Mortality. Jack pine budworm defoliation, if severe enough, can cause considerable mortality. In attempting to simulate this problem the following characteristics of budworm caused mortality were noted:

1. Although mortality in older stands is usually much lower, up to one-third of a stand can be killed if defoliation is severe.

2. Young understory trees can suffer as much as 90 percent mortality.

3. Mortality is more likely to occur to stands severely attacked for more than one year than to stands with only one year's defoliation (Graham, 1935; Benjamin et al., 1961; Dixon, 1961; Kulman et al., 1963; Batzer and Millers, 1970).

To simulate these characteristics a function would be needed in which generated mortality levels usually would be low but which would have the potential to become high if budworm damage was severe over several years. The functions from Table 5 were chosen to produce these results. Only when the sum of defoliation levels over the last three years is over 20 do mortality levels start to climb. Mortalities of 90 and 33 percent for the younger and older stands, though unlikely, are possible. For most years, mortality will be moderate. Before the age of 20, the mortality level determined from the model is used to adjust the number of living trees per acre and eventually will affect the basal area which is initialized at year 20. After age 20, the mortality level obtained is used to adjust the number of living trees and the volume of wood per acre.

Loss Of Merchantability. The model used to simulate white pine weevil damage on jack pine trees is essentially the same as the one proposed by Sapio et al. (1985). Each tree is allowed to be hit four times by the weevil before it is culled. The model works by keeping track of what proportion of the stand has been hit zero, one, two, three, and four times. Assuming that the weevil shows no preference for unattacked trees, these values can be adjusted by the

Table 5. The relationship between three year cumulative budworm population values and percent jack pine mortality.

Sum of budworm population values over the last three years	Percent mortality for stands younger than 20 years	Percent mortality for stands older than 20 years
0	1	1
1	1	1
2	1	1
3	1	1
20	1	1
21	3	2
22	8	5
23	17	8
24	32	14
25	56	22
26	90	33

For stands < 20 years old: $Y = (0.1096X - 1.8764)^4$
For stands > 20 years old: $Y = (0.0736X - 1.1561)^4$

Where Y = Percent mortality of jack pine this year.
X = Sum of budworm population values over the last three years.

proportion of trees hit by the weevil that year. This model is applied to the stand until the trees are 17 feet tall. When this happens, at least 17 feet of straight stem will be available to the logger and the weevil will no longer be an important consideration.

Consider a case where the proportion of trees not hit at the beginning of the year is 0.5, the proportion of trees hit once is 0.2, and the proportion of trees hit by weevils that year is 0.2 (Figure 3). The proportion of unattacked trees which will remain that way will be 40 percent (0.8 x 0.5 = 0.4). The proportion of unattacked trees which will move to the "attacked once" class will be 10 percent (0.2 x 0.5 = 0.1). The proportion of trees hit once at the beginning of the year which will remain that way will be 16 percent (0.8 x 0.2) = 0.16). Therefore, 40 percent of the trees will not be hit at the end of the year, and 26 percent will be hit once (0.1 x 0.26 = 0.26).

The values thus obtained give the proportion of trees culled by the weevil. Since tree DBH values do not differentiate significantly during the critical period, it is assumed that this more or less equals the proportion of wood volume culled by the insect. When the stand is cut, the final wood volume obtained is adjusted by this value.

Modeling Jack Pine Growth

Effects of Site Preparation and Planting. There are many ways to produce a stand of jack pine. Silvicultural systems include clearcutting, seed tree, and shelterwood methods. For each of these, reproductive success will depend, among other things, on what source is used for seeds or seedlings and on what type of site preparation is used. To include all possible options would make the

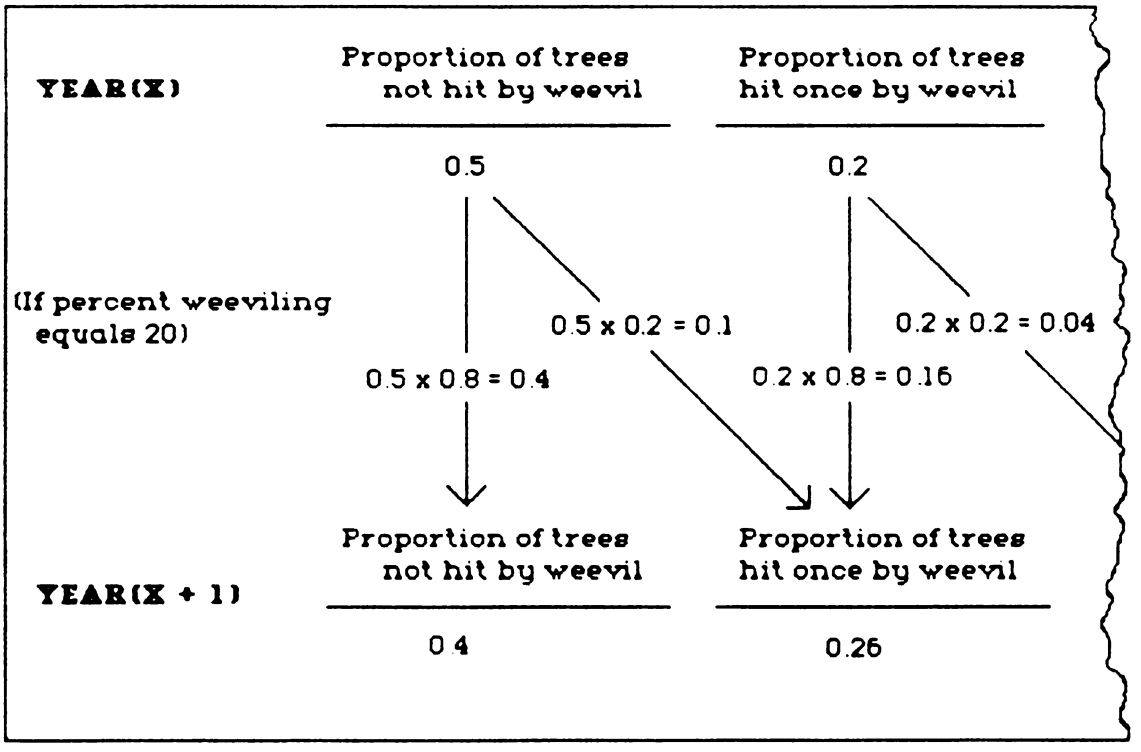


FIGURE 3. Example of how weevil damage on jack pine is up-dated in the model.

model tremendously unwieldy and would obscure insect management, which is the main focus. Therefore, site preparation options were limited to five: burning, scarification, rollerchopping, lop and scattering, and applying a herbicide. Only clearcutting was used as a silvicultural system, and within this system regeneration was limited to five options: natural regeneration, aerial seeding, seeding with a corn planter, planting by hand, and machine planting. No precise numerical information was available on the regeneration success of each option, but enough information was available to get a general idea of what could be expected (Zehngraff, 1943; Cayford, 1958, 1959; Chrosciewicz, 1959, 1960; Hacker et al., 1983).

Even with this simplified set of site preparation and planting options, the number of possible combinations are astronomical; therefore, only a very limited subset of these combinations was simulated. An evenly distributed random number was used to obtain the initial number of trees growing per acre. The upper and lower values for the distribution was set for each combination considered by the upper and lower values historically observed in stands prepared that way. Table 6 shows all site preparation and planting combinations considered and the resulting range of trees per acre.

Note that for a combination of a burn and natural regeneration that two even distributions are used, one with a 75 percent chance of being applied and the other with a 25 percent chance. This was done to simulate the fact that when natural regeneration after a burn is tried anything from zero to 10,000 trees per acre can come up, but that inadequate regeneration is more likely than excessive regeneration.

TABLE 6. The number of trees per acre at stand age two given the site preparation and planting method used.

Planting Method	Range of Possible Site Preparation	Values for Regeneration
Natural Regeneration	Burn	000 - 900 with 75% probability 900 - 10,000 with 25% probability
	Lop and Scatter + Scarification	300 - 900
	Rollerchop	100 - 900
	None of the Above	000 - 400
Seed With a Corn Planter	Burn or Scarify or Herbicide or Rollerchop	100 - 900
	None of the Above	000 - 600
Aerial Seed	Burn or Scarify or Herbicide or Rollerchop	000 - 800
	None of the Above	000 - 500
Hand Plant		900 - 900
Machine Plant		900 - 900

Generating Initial Basal Area. Initial basal area is set when the stand reaches year 20. At that time, a value for the number of trees per acre, a site index, and some indication of what proportion of normal growth has occurred over the last 20 years are available from the model. This information is then used to obtain an initial basal area from Table 7 (Benzie, 1977). The number of trees is cross-referenced with the site index to obtain a basal area value which then is adjusted by the proportion of normal growth. Code for the model was written so that basal area values could be interpolated for those values of site index and tree number that did not exactly fit those on the chart. The computer code was also written so that basal area values could be extrapolated from the chart where high mortality left the stand with fewer than 500 trees per acre at the age of 20. If zero trees per acre were growing on any of the listed site indices, then a basal area of zero was assumed to result. A Lagrange interpolation polynomial was used to connect those zero-tree points with the 500-tree points listed on the chart (Manetsch and Park, 1980).

Growth of the Stand After Age Twenty. After age 20, the increment in basal area is calculated each year (Benzie, 1977):

$$\text{Basal area growth} = 0.276\,(\text{site index})^{0.63}\,\text{Exp}\,(-4\,\text{Exp}\,(-29.18/\text{age}))\,(1 + 0.056\,(\text{basal area}) - 0.000358\,(\text{basal area})^2)$$

This value is then adjusted to account for losses due to budworm damage and is added to the total basal area for the stand. Stand height is calculated (Benzie, 1977):

TABLE 7. Estimated basal area per acre in 20-year-old jack pine stands. (Taken from Benzie, 1977)

Site index	Number of trees per acre			
	500	1000	1500	2000
feet	square feet			
60	39	63	79	92
50	28	46	55	63
40	17	29	36	39

$$\text{Height} = 1.633 \times (\text{site index}) (1 - e^{-0.02233 (\text{age})})^{1.2419}$$

The total basal area is then adjusted to account for mortality losses due to the budworm. The number of cords of wood is calculated using the following equation taken from Benzie (1977):

$$\text{Cords} = 0.003958 \times \text{basal area} \times \text{height}$$

The number of cubic feet of wood is calculated (Benzie, 1977):

$$\text{cubic feet} = 0.4085 \times \text{basal area} \times \text{height}$$

Economic Considerations

Every management activity has a cost associated with it. Each time the stand is managed, this cost and the year during which the activity is performed are recorded. The cost is then discounted back to the year of stand initialization using the following formula:

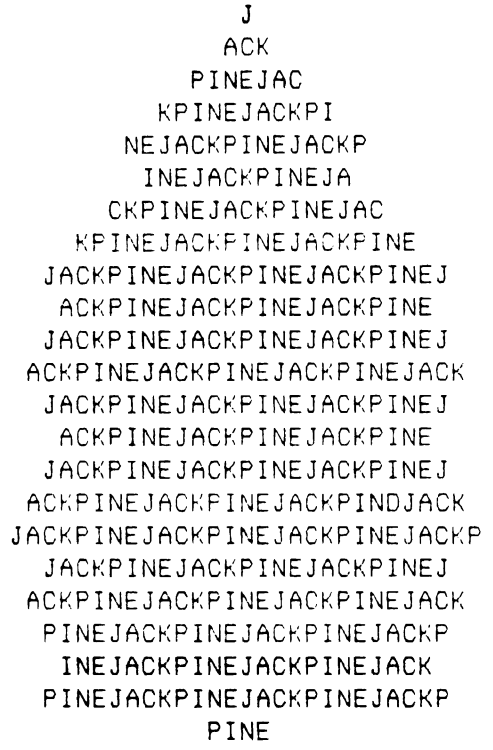
$$\text{Discounted Cost} = \text{Cost} / (1 + i)^n$$

where i = discount rate/100 and n = number of years discounted. This discounting process equalizes monetary values over time. Future money is not worth as much as money held in the present since that same sum can be invested and it will gain interest. This discounting process is very important for any long term investment in a stand of timber. The discount rate must reflect the interest rate

that could potentially be gotten from an alternative investment. After one rotation of the stand, the discounted costs and benefits can be compared to see how well the investment in jack pine compares with alternative investments. If the net sum of benefits and costs is negative then the money would have been better spent elsewhere. A positive result indicates that the investment compares favorably to the alternatives.

Final Evaluation of Insect Monitor and Control Program

Management activities are evaluated two ways in the coded version of the model. First, all discounted costs and benefits are totalled for the entire rotation. The net present value thus obtained tells whether the overall management practiced on the stand was economically viable. Second, a net present value is determined for all monitor and control activities. A second unmanaged stand of jack pine is grown throughout the rotation concurrently with the managed stand. This control is used to determine how much wood was saved by insect control efforts. The discounted value of this wood then is summed with the discounted costs associated with those efforts. This summation yields a net present value for insect control activities for the year of stand initiation. As in the total net present value, a positive value indicates that the investment in control activities compares favorable with alternative investments.



A USERS' MANUAL FOR

PROGRAM BANKSIANA

INTRODUCTION

BANKSIANA is a menu-driven program written in p-system pascal for the IBM Computer. Its purpose is to simulate the growth of jack pine stands under different methods of management for insect control and to evaluate those different methods economically. The insects modeled include the jack pine budworm and the white pine weevil. Control alternatives are limited to determining the timing of sprays and deciding the type of spray operation to implement. Insect monitoring activities are included in the model as an aid in deciding when to spray.

At the beginning of the program, the user can initiate a stand by any one of several site preparation and planting methods. Once a stand is initiated, the program enters the main phase where tree growth under insect attack is simulated. During this phase, insect monitoring and control decisions are made. At the end of a program run, all costs associated with control efforts are discounted to the year of stand initiation and are compared with the discounted value of wood saved to yield a net present value for control efforts.

Stand growth is simulated by equations developed by Paul Laidly and published in the Manager's Handbook for Jack Pine in the North Central States (Benzie, 1977). Weevil damage is simulated similar to Sapio et al. (1985). Jack pine budworm population dynamics are simulated in a similar to Nyrop et al. (1983) and budworm impact is based on observations made by Kulman et al. (1963), Batzer and Millers (1970), Graham (1935), Benjamin (1952, 1953, 1954), Benjamin et al. (1961), and Dixon (1961).

RUNNING BANKSIANA

To run BANKSIANA, you must have an IBM PC or a compatible machine. The program is packaged in a single diskette and only requires one disk drive and a monitor. The following steps are required to run the program:

1) Put the diskette in the disk drive.

2) Start the machine.

3) When the command line appears, type an 'X'.

4) The machine will print 'Execute what file?', respond with '#4:C' and a carriage return.

The program will begin to run and will prompt the user for the required responses.

MENU FORMAT

BANKSIANA is menu-driven; therefore every response required will be listed on the screen at the appropriate times. Two types of responses will be expected: a character response and a numerical response. Character responses will be prompted in the following manner:

'Type of run: R)andom or D)etermined.

In this example, an 'R' will give a random run, 'D' will give a deterministic run. When character responses are required, you do not have to press the return key after entering your response.

Numerical responses require that you press a return key after entering them so that the machine knows when you are finished. Numerical responses are

required either when a numerical value is being assigned to a variable or when numerical values are used to delineate menu choices. An example of a prompt requiring the first type of numerical response is:

Site index (45 - 60): 55

This prompt assigns a value to the variable site index. The values within the parentheses give the range of acceptable values. The number on the right is the default value which the program assigns to the variable. Press the return key to accept this value, otherwise enter a new value which will be printed on the screen over the old default value.

Figure 4 is an example of a prompt requiring the second type of numerical response. When you see this, enter the numerical value corresponding to the item you wish to alter, and then press return. More prompts will be printed on the top of the screen which will steer you through the steps needed to make whatever changes you wish. Available options will be listed on the screen. Finally, if you wish to leave the menu, the last one or two numbers listed will provide an out.

INITIALIZING YOUR STAND

The first prompt to appear on the screen when the program starts will look like this:

> You have the option to change the way certain parts of this program operate. Do you wish to see the menu and make some changes? .. Y or N

```
Which of the items listed below do you want to change?  0
          JACK PINE MANAGEMENT MODEL -- INITIALIZATION PHASE

        1) Type of user:       Public agency

        2) Type of run:      Random

        3) Number of acres (10 - 999):  40

        4) Site index (45 - 60): 55

        5) Discount rate % (0.0 - 15.0):  3.00

        6) Interactive vs. Automatic run.   (Default = interactive)

        7) Initial weevil density: moderate

        8) Initialize management costs

        9) Select site preparation and planting

       10) Run model

       11) Exit program
```

Figure 4. Main initialization menu.

If you choose 'N' a default version of the program will run. This is handy for people who might find the menus intimidating. With 'Y' a menu (Figure 4) will print on the screen. With this menu, a user can alter the program run. Most of the options shown in Figure 4 simply allow you to change the values of some variables. For these, prompts similar to those shown in the previous section will appear at the top ot the menu. Other options will result in another menu being printed on the screen. For these, the responses required may be a little more complicated.

Option number 6 allows a user to decide whether to manage the stand interactively as the program progresses or to set up a management plan at the start and simply let the program run by itself. If the program runs automatically, the menu (Figure 5) will appear on the screen. This menu is a worksheet for setting up a plan for managing a stand. A 'B' or 'W' allows a user to set up a plan for dealing with the budworm or the weevil. Prompts will appear querying the user on whether or not to monitor, what to spray, etc. An 'R' determines how long the stand will grow before cutting. A standard numerical response prompt will be needed for this. To get out of this menu, press a 'G'.

To change some of the costs involved in monitoring and spraying insects press the '8'. The menu shown in Figure 6 will then appear. All of the entries for this menu entail simple numerical assignments and follow the standard format. Two values are associated with spray costs: the initial cost required to proceed with the operation and per acre costs associated with the spray operation. The right value on the menu represents the total cost that will be incurred whenever a spray operation of this type is undertaken. It is determined by the first two values listed on the menu and the total number of acres within your stand.

```
     Budworm? Weevil? Rotation? or Go on to next part?

JACK PINE BUDWORM AND WHITE PINE WEEVIL MONITORING AND CONTROL PLAN
-------------------------------------------------------------------

     B)udworm plan: (Default = ignore budworm)

     W)eevil plan: (Default = ignore weevil)

     R)otation length: 40 years

G)o on to the next part of the program
```

Figure 5. Initialization menu for automatic management in the model.

```
Which of the items listed below do you wish to change?
          PACK PINE MANAGEMENT MODEL -- MANAGEMENT COSTS
-------------------------------------------------------------------------------
  Initialize wages of personnel.
        1) Hourly wage of pest scout:    5.00

        2) Hourly wage of forest technician:    8.00

  Initialize management costs for pesticide application.
     Applications targeted against Jack Pine Budworm
        3) Aerial, chemical spray:    0.00 +    8.00 per acre =    320.00

        4) Ground, chemical spray:    0.00 +   20.00 per acre =    800.00

        5) Aerial, biological spray:  0.00 +   10.00 per acre =    400.00

        6) Ground, biological spray:  0.00 +   25.00 per acre =   1000.00

     Applications targeted against White Pine Weevil
        7) Aerial, chemical spray:    0.00 +    8.00 per acre =    320.00

        8) Ground, chemical spray:    0.00 +   30.00 per acre =   1200.00

     9) Go back to the initialization menu
    10) Exit program
```

Figure 6. Cost setting menu.

```
Which of the items listed below do you want to change?  2

     JACK PINE MANAGEMENT MODEL -- SITE PREPARATION AND REGENERATION

         1) Burn:  No

         2) Herbicide:  No

         3) Lop and scatter:  No

         4) Scarify:  No

         5) Rollerchop:  No

         6) Method of regeneration:              Machine plant


7) Go back to the initialization menu
```

Figure 7. Regeneration and site preparation menu.

As an example of how to use this menu, to change the costs associated with aerially spraying jack pine budworm with chemical pesticide, press the '3' and the return. At this point the same line as that next to the '3' will appear at the top of the menu with the curser on top of the left-most value. Enter the desired number, and hit the return. The curser will then move over to the next value which can be changed. Note that after either value is changed, the right-most value on the menu changes as well. After all desired changes are made, press a '9' to get back to the main menu.

For option number 9 on the main menu, the menu shown in Figure 7 will appear on the screen. This menu sets what type of site preparation and planting will be used to initialize the stand, and it sets the costs that will be associated with those activities. To get back to the main menu, press a '7'.

Once all changes are made, press a '10' to run the program. The program will determine how many trees per acre the stand will start with. Further action may be required if regeneration is too thick or too sparse.

INTERACTIVE MANAGEMENT OF THE STAND

When the program is run interactively, the years of the rotation can be divided between inactive years where nothing is done to manage the stand and active years where management activities ranging from conducting insect sampling to cutting the stand can be practiced. These active years are determined two ways.

1. Starting from year one and for each active year afterwords, decide how many inactive years before the next active year. This is set with the following prompt:

How many years should we wait before inspecting the stand?

2. During inactive years when a budworm outbreak occurs, a user will be able to make the year active. If this happens, the machine will beep and present this prompt:

WARNING! A budworm outbreak is now occurring. You have 3 seconds to act on this. Press any key to halt the program.

By pressing any key on the keyboard within the allotted time, the program will stop and the user will be allowed to actively manage the stand for that year.

Active years can be divided into two phases: insect monitoring and management. During the monitoring phase, which occurs first, the user will be able to sample one or both insects any number of times. When all the sampling is completed, press the letter from the menu corresponding to the option:

GO ON TO MANAGEMENT.

During the management phase, the options are to spray insecticides, inventory the stand, check up on expenditures to date, cut the stand, or leave the program. All or none of these options can be done any number of times, until the user leaves the program or cuts the stand. With either of these choices, the user will exit the monitor/manage portion of the program. To continue growing the stand and quit management, press the letter corresponding to:

DO NOTHING - GO ON TO NEXT YEAR.

The program will continue alternating between active years and inactive years until the user (1) exits the program during a management phase, (2) cuts the stand, or (3) lets the program run past year 70 at which time the stand must be cut. In the first case you will go back to the command line of the p-system operating system where you will have the opportunity to re-enter the program if you desire. In the last two cases you will go onto the performance evaluation portion of the program.

FINAL EVALUATION OF MANAGEMENT ACTIVITIES

Figure 8 is an example of the final output. The top half shows an evaluation of the economic effectiveness of the rotation taken as a whole. All costs incurred over the rotation are discounted to the year of stand initiation and subtracted from the amount gained at harvest, also discounted to the year of initiation. The value obtained (net present value) tells whether or not the money invested into this stand compared favorably with available alternative investment opportunities. The second half of the final evaluation focuses on those activities related to insect control. Costs incurred in monitoring and controlling insect populations are compared similarly to the value of wood saved to determine how the investment in insect control compared with alternative investments. A negative net present value does not necessarily mean that less money was taken in than was spent; it means that more money could have been earned had an alternative investment been chosen.

```
                    EVALUATION OF ALL ACTIVITIES
                    ----------------------------
                                The number of cords cut=   10.8
                       Discounted value for these cords=   24.56
                     Discounted value of costs incurred=  166.66
              Net present value of the stand at planting= -142.10

Price of jack pine which would be required to break even=  47.49


                EVALUATION OF MONITOR AND CONTROL ACTIVITIES
                -------------------------------------------
    Number of cords saved by monitor and control program=    4.5
                       Discounted value of these cords=   10.14
          Discounted cost of monitor and spray program=   86.66
        Net present value of monitor and spray program=  -76.52



                                                        RETURN
```

Figure 8. Final evaluation of management activities chosen.

CONSIDERING TAXES

The results obtained from this program are most applicable to jack pine stands grown by a public agency since taxes were not included. However, the results should still be of value to people from the private sector as long as they are aware of this and make appropriate adjustments for themselves. Taxes generally tend to make investments in forestry more competitive with alternate investments than they might otherwise be (R. J. Marty, 1984, Michigan State University, personal communication); therefore, any set of investments which result in a marginally negative net present value may compare favorably to alternatives after considering taxes.

LIST OF REFERENCES

Banash, S.E. 1958. White pine weevil appraisal survey. USDA Coop. Econ. Rpt. 8:871-872.

Batzer, H.O., and D.T. Jennings. 1980. Numerical analysis of a jack pine budworm outbreak in dense jack pine. Environ. Entomol. 9:514-524.

Benjamin, D.M., S.E. Banash, and R.B. Stewart. 1961. Losses attributable to the jack pine budworm during 1955-1957 outbreaks in Wisconsin. University of Wisconsin Forestry Research Notes No. 73.

Benzie, J.W. 1977. Managers handbook for jack pine in the North Central States. USDA For. Serv. Gen. Tech. Rep. NC-32, 18 p., North Central Forest Expt. Sta., St. Paul, Minnesota.

Belyea, R.M. and C.R. Sullivan. 1956. The white pine weevil: a review of current knowledge. For. Chron. 32:58-67.

Cayford, J.H. 1958. Scarifying for jack pine regeneration in Manitoba. Canada Dept. of Northern Affairs and National Res., For. Br., For. Res. Div. Tech. Note No. 66. 14 p.

Cayford, J.H. 1959. Germination and survival of jack pine and red pine after scarification in Southeastern Manitoba. Canada Dept. of Northern Affairs and National Res., For. Br., For. Res. Div. Tech. Note No. 79. 14 p.

Chrowsciewicz, Z. 1959. Controlled burning experiments on jack pine sites. Canada Dept. Northern Affairs and National Res., For. Br., For. Res. Div. Tech. Note No. 72. 19 pp.

Chrowsciewicz, Z. 1960. Jack pine regeneration after scattering slash on exposed mineral soil. Pulp and Paper Mag. of Canada, Woodlands Review 61:164-166.

Dixon, J.C. 1981. Report on project to develop a survey manual for jack pine budworm. Dept. of Biology. University of Wisconsin, Eau Claire, Wisconsin.

Foltz, J.L., Knight F.B., and D.C. Allen. 1972. Numerical analysis of population fluctuations of the jack pine budworm. Ann. Entomol. Soc. Amer. 65:82-89.

Fowells, H.A. 1965. Silvics of forest trees of the United States. USDA Timber Management Research. Forest Service. Agricultural Handbook No. 271. 762 p.

Graham, S.A. 1935. The spruce budworm on Michigan pine. Univ. Mich., School For. and Conservation Bull. 6.

Hacker, J.J., P.E. Marshall, A.E. Erickson. 1983. A review of jack pine regeneration in the Lake States. Michigan State University Agric. Expt. Sta. Research Rept 443. 31 p.

Krebs C.F. 1972. A procedure for sampling jack pine damaged by the white pine weevil and several notes on the insect's effect on stand de-velopement. Phd Dissertation. University of Michigan. Ann Arbor, Michigan. 119 p.

Kulman, H.M., A.C. Hodson, D.P. Duncan. 1963. Distribution and effects of jack pine budworm defoliation. Forest Sci. 9:146-157.

Manetsch, T.J. 1980. Systems analysis and simulation with application to economic and social systems. Part II. Michigan State University, East Lansing, Michigan. 239 p.

Marty, R.J. and D.G. Mott. 1964. Evaluating and scheduling white pine weevil control in the Northeast. USDA For. Serv. Res. Paper. NE-19. 56 p.

Morse F.S. 1958. The white pine weevil in Wisconsin jack pine plantations. Masters Thesis. University of Wisconsin. 44 p.

Nyrop, J.P., J.T. Olson, D.G. Mosher, and G.A. Simmons. 1983. Simulation of how jack pine budworm affects economic returns from jack pine timber production in Michigan. 16:157-165.

Sapio, F.J., R.J. Drapek, G.A. Simmons, and R.L. Murray. 1985. Prediction of jack pine volume losses and plantation decline caused by the white pine weevil. Northern J. Appl. For. (in press).

Zehngraff, P.J. 1943. Jack pine regeneration following clearcutting on the Chippewa National Forest. J. For. 41:122-125.

APPENDIX A

A Listing of the Program

```
(*$s+++*)
PROGRAM BANKSIANA;
USES SCREENOPS, (*$U RD.LIBRARY *) UTILITIES,RANDOM_NUMBER_GENERATORS;
(*$I DISK3:DECLAR.TEXT *)
(*$I DISK3:AUTO.TEXT *)
(*$I DISK3:COST.TEXT *)
(*$I DISK3:PREP.TEXT *)
(*$I DISK3:INITBIOL.TEXT *)
(*$I DISK3:INIT.TEXT *)
(*$I DISK3:SPRAY.TEXT *)
(*$I DISK3:MANAGE.TEXT *)
(*$I DISK3:MONITOR.TEXT *)
(*$I DISK3:SETPEST.TEXT *)
(*$I DISK3:MAIN.TEXT *)
END.
```

```
(*****************  DECLAR.TEXT  ******************)

TYPE
  INSECT_TYPE = (BUDWORM,WEEVIL);
  METHOD_TYPE = (AERIAL,GROUND);
  SPRAY_TYPE = (CHEMICAL,BIOLOGICAL);
  FUNC_ROLE = (SLOPE,INTERCEPT);

VAR
  ACTIVITY:STRING[20];     (* Holds description of each mgmnt activity *)
  AIR_SEED_COST:REAL;      (* Cost per acre to spread seed by air      *)
  AUTO:BOOLEAN;            (* Determines run interactive or automatic  *)
  BA_W_CONTROL:REAL;       (* Basal area of stand with control program *)
  BA_WO_CONTROL:REAL;      (* B. A. of stand with no control program   *)
  BID:REAL;                (* Monetary offer given to cut the stand    *)
  BUD_COUNT:INTEGER;       (* Determines when to sample or manage JPBW *)
  BUD_TABLE:ARRAY [1..5,1..3] OF REAL;
                           (* Table holding growth loss proportions    *)
  BURN:BOOLEAN;            (* Determines if burn was used in site prep *)
  BURN_COST:REAL;          (* Cost per acre to burn                    *)
  BW1_WO_CONTROL:REAL;     (* Budworm this yr in stand without control *)
  BW1_W_CONTROL:REAL;      (* Budworm this yr in stand with control    *)
  BW2_WO_CONTROL:REAL;     (* Budworm last yr in stand without control *)
  BW2_W_CONTROL:REAL;      (* Budworm last yr in stand with control    *)
  BW3_WO_CONTROL:REAL;     (* Bworm 2 yrs ago in stand without control *)
  BW3_W_CONTROL:REAL;      (* Budworm 2 yrs ago in stand with control  *)
  BW_CHECK:INTEGER;        (* In auto, determines when to check bworm  *)
  BW_MON_PLAN:STRING[14]; (* Describes plan in auto to monitor bw      *)
  BW_PLAN:STRING[8];       (* Describes general plan for bw in auto    *)
  BW_SPRAY_TYPE:CHAR;      (* Determines in auto choice of spray, bw   *)
  CP_SEED_COST:REAL;       (* Cost per acre to seed with corn planter  *)
  CH:CHAR;                 (* Used, various parts of program for input *)
  CHECK:INTEGER;           (* Num yrs to stand inspection, interactive *)
  CHOP_COST:REAL;          (* Cost per acre to run a roller chopper    *)
  CONTROL_COSTS:REAL;      (* Summation of control costs discounted    *)
  COSTS:REAL;              (* Summation of all costs discounted        *)
  COSTSIN:TEXT;            (* Text file retain costs for future info   *)
  COSTSOUT:TEXT;           (* 2nd file as above allows manipulations   *)
  COUNTYR:INTEGER;         (* Years since initial year                 *)
  CORDS_W_CONTROL:REAL;    (* Cords in stand with control measures     *)
  CORDS_WO_CONTROL:REAL;   (* Cords in stand with no control measures  *)
  CUT:BOOLEAN;             (* Determines if the stand has been cut     *)
  DISCOUNT:REAL;           (* Discount rate used                       *)
  END_BW_PLAN:INTEGER;     (* In auto, yr to stop bw related activities*)
  END_WV_PLAN:INTEGER;     (* In auto, yr to stop wv related activities*)
  FIRST_WEEV:REAL;         (* Initial weevil level                     *)
  HEIGHT:REAL;             (* Tree height                              *)
  HERB_COST:REAL;          (* Cost per acre to apply a herbicide       *)
  HERBICIDE:BOOLEAN;       (* Determines if herbicide was used         *)
  HIT_0X_W_CONTROL:REAL;   (* Prop. trees not hit by uncontrolled wv   *)
  HIT_0X_WO_CONTROL:REAL;  (* Prop. trees not hit by controlled weevil *)
  HIT_1X_WO_CONTROL:REAL;  (* Prop. trees hit 1X by uncontrolled wv    *)
  HIT_2X_WO_CONTROL:REAL;  (* Prop. trees hit 2X by uncontrolled weevil*)
  HIT_3X_WO_CONTROL:REAL;  (* Prop. trees hit 3X by uncontrolled weevil*)
```

```
  HIT_4X_WO_CONTROL:REAL; (* Prop. trees hit 4X by uncontrolled weevil*)
  HIT_1X_W_CONTROL:REAL;  (* Prop. trees hit 1X by controlled weevil  *)
  HIT_2X_W_CONTROL:REAL;  (* Prop. trees hit 2X by controlled weevil  *)
  HIT_3X_W_CONTROL:REAL;  (* Prop. trees hit 3X by controlled weevil  *)
  HIT_4X_W_CONTROL:REAL;  (* Prop. trees hit 4X by controlled weevil  *)
  HND_PLANT_COST:REAL;    (* Cost per acre to plant by hand           *)
  IS_IT_CONTROL:BOOLEAN;  (* Used to determine if cost is control cost*)
  LASTBUDWORM:INTEGER;    (* Years since last budworm outbreak        *)
  LASTWEEV:REAL;          (* Weevil level which will occur yr 20      *)
  LG_LARV_LEV:REAL;       (* In auto, lge bw sample merits more action*)
  LOP_SCATTER:BOOLEAN;    (* Determines if a lop and scatter was done *)
  MAC_PLANT_COST:REAL;    (* Cost per acre to machine plant           *)
  NEXTBUDWORM:INTEGER;    (* Years to next budworm outbreak           *)
  N_TRS_W_CONTROL:REAL;   (* Num trees in stand being controlled      *)
  N_TRS_WO_CONTROL:REAL;  (* Num trees in stand not being controlled  *)
  NUMACRES:INTEGER.       (* Holds the number of acres in the stand   *)
  NUMEXPENDITURES:INTEGER;(* Number of expenditures undertaken        *)
  PLANT_METHOD:STRING[24];(* Nat regen vs seeding vs planting         *)
  PRICE:REAL;             (* Cost of each management activity         *)
  P_GRO_W_CONTROL:REAL;   (* Proportion potential growth with control *)
  P_GRO_WO_CONTROL:REAL;  (* Prop. potential growth with no control   *)
  P_LIVE_W_CONTROL:REAL;  (* Prop. original trees alive with control  *)
  P_LIVE_WO_CONTROL:REAL; (* Prop. original trees alive no control    *)
  PRESENT_VALUE:REAL;     (* Cost of any activity at time it is done  *)
  QUANTITY:REAL;          (* Number of units of mgmt activity done    *)
  RAN_OR_DET:STRING[10];  (* Random or determined run                 *)
  READIN:STRING[16];      (* File name to be used to read costs into  *)
  READOUT:STRING[16];     (* File name to be used to read costs out of*)
  ROLLERCHOP:BOOLEAN;     (* Determines if rollerchop was used        *)
  ROTATION:INTEGER;       (* In auto, determines rotation length      *)
  SAMP_LARGE:BOOLEAN;     (* In auto, determines if sample lrge bw    *)
  SCARIFY:BOOLEAN;        (* Notes if scarification was done          *)
  SCAR_COST:REAL;         (* Cost per acre to scarify                 *)
  SCOUT_WAGE:REAL;        (* Wage of pest scout                       *)
  SITEINDEX:INTEGER;      (* Stand site index                         *)
  SM_LARV_LEV:REAL;       (* In auto, sample level small  bw to go on *)
  SPRAY_BUD:BOOLEAN;      (*                                          *)
  SPRAY_COST:ARRAY [BUDWORM..WEEVIL, AERIAL..GROUND, CHEMICAL..BIOLOGICAL,
                    SLOPE..INTERCEPT] OF REAL;
                          (* Costs associated with pesticide sprays   *)
SPRAY_WV:BOOLEAN;        (* Notes whether weevil has been sprayed    *)
 START_BW_PLAN:INTEGER;  (* If in auto: year to start budworm plan   *)
 START_WV_PLAN:INTEGER;  (* If in auto: year to start weevil plan    *)
 TECH_WAGE:REAL;         (* Wage of forest technician                *)
 TOGGLE:BOOLEAN;         (* Used for cost file manipulation          *)
 TREE_AGE:INTEGER;       (* How old the trees are.                   *)
 USER:STRING[18];        (* Point of view of user ... Not Used!      *)
 VALUE_DISCOUNTED:REAL;  (* Used to hold values of costs discounted  *)
 WEEV_CHECK:INTEGER;     (* If in auto: num yrs between weevil checks*)
 WEEV_COUNT:INTEGER;     (* If auto: to see when weev check reached  *)
 WEEV_PLAN:STRING[8];    (* Auto: how weevil will be dealt with      *)
 WEEV_SPRAY:BOOLEAN;     (* To show that weevil has been sprayed     *)
 WEEV_LEVEL:STRING[8];   (* Low moderate high or extreme             *)
 WPW_W_CONTROL:REAL;     (* Weevil level on controlled stands        *)
```

```
  HIT_4X_WO_CONTROL:REAL; (* Prop. trees hit 4X by uncontrolled weevil*)
  HIT_1X_W_CONTROL:REAL;  (* Prop. trees hit 1X by controlled weevil  *)
  HIT_2X_W_CONTROL:REAL;  (* Prop. trees hit 2X by controlled weevil  *)
  HIT_3X_W_CONTROL:REAL;  (* Prop. trees hit 3X by controlled weevil  *)
  HIT_4X_W_CONTROL:REAL;  (* Prop. trees hit 4X by controlled weevil  *)
  HND_PLANT_COST:REAL;    (* Cost per acre to plant by hand           *)
  IS_IT_CONTROL:BOOLEAN;  (* Used to determine if cost is control cost*)
  LASTBUDWORM:INTEGER;    (* Years since last budworm outbreak        *)
  LASTWEEV:REAL;          (* Weevil level which will occur yr 20      *)
  LG_LARV_LEV:REAL;       (* In auto, lge bw sample merits more action*)
  LOP_SCATTER:BOOLEAN;    (* Determines if a lop and scatter was done *)
  MAC_PLANT_COST:REAL;    (* Cost per acre to machine plant           *)
  NEXTBUDWORM:INTEGER;    (* Years to next budworm outbreak           *)
  N_TRS_W_CONTROL:REAL;   (* Num trees in stand being controlled      *)
  N_TRS_WO_CONTROL:REAL;  (* Num trees in stand not being controlled  *)
  NUMACRES:INTEGER;       (* Holds the number of acres in the stand   *)
  NUMEXPENDITURES:INTEGER;(* Number of expenditures undertaken        *)
  PLANT_METHOD:STRING[24];(* Nat regen vs seeding vs planting         *)
  PRICE:REAL;             (* Cost of each management activity         *)
  P_GRO_W_CONTROL:REAL;   (* Proportion potential growth with control *)
  P_GRO_WO_CONTROL:REAL;  (* Prop. potential growth with no control   *)
  P_LIVE_W_CONTROL:REAL;  (* Prop. original trees alive with control  *)
  P_LIVE_WO_CONTROL:REAL; (* Prop. original trees alive no control    *)
  PRESENT_VALUE:REAL;     (* Cost of any activity at time it is done  *)
  QUANTITY:REAL;          (* Number of units of mgmt activity done    *)
  RAN_OR_DET:STRING[10];  (* Random or determined run                 *)
  READIN:STRING[16];      (* File name to be used to read costs into  *)
  READOUT:STRING[16];     (* File name to be used to read costs out of*)
  ROLLERCHOP:BOOLEAN;     (* Determines if rollerchop was used        *)
  ROTATION:INTEGER;       (* In auto, determines rotation length      *)
  SAMP_LARGE:BOOLEAN;     (* In auto, determines if sample lrge bw    *)
  SCARIFY:BOOLEAN;        (* Notes if scarification was done          *)
  SCAR_COST:REAL;         (* Cost per acre to scarify                 *)
  SCOUT_WAGE:REAL;        (* Wage of pest scout                       *)
  SITEINDEX:INTEGER;      (* Stand site index                         *)
  SM_LARV_LEV:REAL;       (* In auto, sample level small  bw to go on *)
  SPRAY_BUD:BOOLEAN;      (*                                          *)
  SPRAY_COST:ARRAY [BUDWORM..WEEVIL, AERIAL..GROUND, CHEMICAL..BIOLOGICAL,
                    SLOPE..INTERCEPT] OF REAL:
                          (* Costs associated with pesticide sprays   *)
SPRAY_WV:BOOLEAN;        (* Notes whether weevil has been sprayed    *)
  START_BW_PLAN:INTEGER;  (* If in auto: year to start budworm plan   *)
  START_WV_PLAN:INTEGER;  (* If in auto: year to start weevil plan    *)
  TECH_WAGE:REAL;         (* Wage of forest technician                *)
  TOGGLE:BOOLEAN;         (* Used for cost file manipulation          *)
  TREE_AGE:INTEGER;       (* How old the trees are.                   *)
  USER:STRING[18];        (* Point of view of user ... Not Used!      *)
  VALUE_DISCOUNTED:REAL;  (* Used to hold values of costs discounted  *)
  WEEV_CHECK:INTEGER;     (* If in auto: num yrs between weevil checks*)
  WEEV_COUNT:INTEGER;     (* If auto: to see when weev check reached  *)
  WEEV_PLAN:STRING[8];    (* Auto: how weevil will be dealt with      *)
  WEEV_SPRAY:BOOLEAN;     (* To show that weevil has been sprayed     *)
  WEEV_LEVEL:STRING[8];   (* Low moderate high or extreme             *)
  WPW_W_CONTROL:REAL;     (* Weevil level on controlled stands        *)
```

```
WPW_WO_CONTROL:REAL;      (* Weevil levil on uncontrolled stands      *)
WV_LEV:REAL;              (* Auto: what weevil level before will spray*)
WV_SPRAY_TYPE:CHAR;       (* Auto: pesticide application for weevil   *)
YEAR_OF_OUTBREAK:INTEGER;(* Which year into budworm outbreak this is*)
YEAR_OF_TRANSACTION:INTEGER;(* Year of each maragement activity      *)
```

```
(* ********** AUTO.TEXT ************ *)

SEGMENT PROCEDURE AUTO_CHOICE;
(*************************************************************************
 * -Called from WELCOME.                                                 *
 * -This procedure determines whether the program will run               *
 *  interactively or automatically.                                      *
 *************************************************************************)
 VAR
   CH:CHAR;
   OK:BOOLEAN;

 PROCEDURE SET_AUTO;
 (************************************************************************
  * -Called from AUTO_CHOICE.                                           *
  * -Sets up program for dealing with budworm and weevil, and           *
  *  determines rotation length if run will be automatic.               *
  ************************************************************************)
  VAR
    CH,CH1,CH2,CH3:CHAR;

  PROCEDURE SET_BW_MON_PLAN;
  (***********************************************************************
   * -Called from SET_BW_PLAN.                                          *
   * -If a monitor and spray program is chosen for budworm; sets        *
   *  up actual monitoring schedule and determines when to go on        *
   *  and spray for budworm.                                            *
   ***********************************************************************)
   BEGIN
     BW_PLAN:='MonSpray';
     FILL_FIELD(25,5,48,' ');
     GOTOXY(25,5);WRITELN('Monitor and Spray');
     CH2:=CH_PROMPT(15,6,CONCAT('How will we monitor for budworm?$',
                    '     E)very ?? years$     O)nly during ',
                    'outbreaks'),'$',48,6,['E','e','O','o']);
     CAP(CH2);
     FILL_FIELD(15,6,35,' ');
     FILL_FIELD(20,7,15,' ');
     FILL_FIELD(20,8,25,' ');
     CASE CH2 OF
       'E':BEGIN
         BW_MON_PLAN:='Every ?? yrs';
         INT_PROMPT(15,7,'How many years between samplings?',
                '$',48,7,BW_CHECK,2);
         FILL_FIELD(15,7,40,' ');
         END;
       'O':BW_MON_PLAN:='During outbrks';
       END; (* Case *)
     CH2:=CH_PROMPT(15,7,CONCAT('Will we sample large larvae as well ',
                'as over-wintering larvae?'),'$',77,7,
                ['Y','','N','yn']);
     CAP(CH2);
     IF (CH2='Y') HEN BEGIN
       SAMP_LARGE:=TRUTE;
```

```
      REAL_PROMPT(15,7,CONCAT('If budworm counts from the over-',
           'winter larval survey$are higher than ?.?, then go on',
           ' and do a late$larval survey.$     Enter a value.$',
           '     (NOTE:Sample values in the program will range$',
           '        from 0.0 - 7.0 larvae per 18 inch branch)'),
           '$',70,12,SM_LARV_LEV,3,1);
      FILL_FIELD(15,7,459,' ');
      REAL_PROMPT(15,7,CONCAT('If budworm counts from the large',
           ' larvae survey are higher$than ??.?, then apply',
           ' control measures.$     Enter a value.$',
           '     (NOTE:Sample values in the program will range$',
           '       from 0.0 - 10.0 larvae per 15 inch branch )'),
           '$',70,12,LG_LARV_LEV,3,1);
      FILL_FIELD(15,7,460,' ');
      END
    ELSE BEGIN
      SAMP_LARGE:=FALSE;
      REAL_PROMPT(15,7,CONCAT('If budworm counts from the over-',
           'winter larval survey$are higher than ?.?, then apply',
           ' control measures.$     Enter a value.$',
           '     (NOTE:Sample values in the program will range$',
           '       from 0.0 - 7.0 larvae per 18 inch branch )'),
           '$',70,12,SM_LARV_LEV,3,1);
      FILL_FIELD(15,7,460,' ');
      END;
    END;                    (* bw mon plan *)

PROCEDURE SET_BW_SPRAY_TYPE;
(*****************************************************************
 * -Called from SET_BW_PLAN.                                     *
 * -Determines type of pesticide to be used against budworm.     *
 *****************************************************************)
  BEGIN
    CH2:=CH_PROMPT(15,6,CONCAT('Choose from below the type of spray ',
         'operation which will be used.$A) Aerial application / ',
         'Chemical pesticide$B) Aerial application / Biological ',
         'pesticide$C) Ground application / Chemical pesticide$',
         'D) Ground application / Biological pesticide'),'$',75,10,
         ['A','a','B','b','C','c','D','d']);
    CAP(CH2);
    BW_SPRAY_TYPE:=CH2;
    FILL_FIELD(15,6,65,' ');
    FILL_FIELD(15,7,60,' ');
    FILL_FIELD(15,8,66,' ');
    FILL_FIELD(15,9,66,' ');
    FILL_FIELD(15,10,66,' ');
    END;                (* of set bw spray type *)

PROCEDURE SET_YRS_BW_PLAN;
(*****************************************************************
 * -Called from SET_BW_PLAN.                                     *
 * -Determines what years to start and stop budworm plan.        *
 *****************************************************************)
  BEGIN
```

```
    REPEAT
      INT_PROMPT(15,7,'What year will we start this plan from?'
         ,'$',55,7,START_BW_PLAN,2);
      UNTIL (START_BW_PLAN >=0 ) AND (START_BW_PLAN < 70);
    REPEAT
      INT_PROMPT(15,7,'What year will we end this plan?'
           ,'$',55,7,END_BW_PLAN,2);
      UNTIL (END_BW_PLAN > START_BW_PLAN) AND (END_BW_PLAN <= 70);
    FILL_FIELD(15,7,65,' ');
    END;                        (* of set bw plan yrs *)

PROCEDURE BW_PLAN_WRITEUP;
(*****************************************************************
 * -Called from SET_BW_PLAN                                      *
 * -Writes out budworm management plan on the screen.            *
 *****************************************************************)
  BEGIN
    IF (CH1 = 'M') THEN BEGIN
      GOTOXY(15,6);
      WRITELN(BW_MON_PLAN);
      IF (BW_MON_PLAN = 'Every ?? yrs') THEN BEGIN
        FILL_FIELD(21,6,2,' ');
        GOTOXY(21,6);
        WRITELN(BW_CHECK);
        END;
      IF SAM_LARPGETHEN BEGI N
        GOTOXY(15,7);
        WRITELN('If more than ',SM_LARV_LEV:3:1,' in winter survey ',
                  'go on.');
        GOTOXY(15,8);
        WRITELN('If more than ',LG_LARV_LEV:4:1,' in late survey then ',
             'spray');
        GOTOXY(15,10);
        WRITELN('Start this program from year ',START_BW_PLAN:2,', and ',
             'end at year ',END_BW_PLAN:2);
        END
      ELSE BEGIN
        GOTOXY(15,7);
        WRITELN('If more than ',SM_LARV_LEV:3:1,' in winter survey ');
        GOTOXY(15,8);
        WRITELN('. . . then spray');
        GOTOXY(15,1);
        WRITELN('Start this program from year ',START_BW_PLAN:2,', and ',
             'end at year ',END_BW_PLAN:2);
        END;
      END
    ELSE BEGIN
      GOTOXY(15,7);
      WRITELN('Starting from year ',START_BW_PLAN:2,' and ending at year ',
           END_BW_PLAN:2);
      END;
    CASE BW_SPRAY_TYPE OF
      'A':IF (CH1='S') THEN
          WRITE_STRING(15,6,'Aerial application/Chemical pesticide',37)
```

```
        ELSE
          WRITE_STRING(15,9,'Aerial application/Chemical pesticide',37);
      'B':IF (CH1='S') THEN
          WRITE_STRING(15,6,'Aerial application/Biological pesticide',39)
        ELSE
          WRITE_STRING(15,9,'Aerial application/Biological pesticide',39);
      'C':IF (CH1='S') THEN
          WRITE_STRING(15,6,'Ground application/Chemical pesticide',37)
        ELSE
          WRITE_STRING(15,6,'Ground application/Chemical pesticide',37);
      'D':IF (CH1='S') THEN
          WRITE_STRING(15,6,'Ground application/Biological pesticide',39)
        ELSE
          WRITE_STRING(15,6,'Ground application/Biological pesticide',39);
      END; (* Case *)
    END;            (* of bw plan writeup *)

PROCEDURE SET_BW_PLAN;
(*****************************************************************
 * -Called from SET_AUTO                                         *
 * -Sets up a plan to manage the stand for jack pine budworm     *
 *****************************************************************)
 BEGIN
   FILL_FIELD(1,1,80,' ');
   WRITE_STRING(1,1,'Which way will you handle budworm?',34);
   FILL_FIELD(25,5,455,' ');
   CH1:=CH_PROMPT(25,5,CONCAT('I)gnore them, S)pray only, or ',
      'M)onitor and spray'),'$',37,1,['I','i','S','s','M','m']);
   CAP(CH1);
   CASE CH1 OF
     'I':BEGIN
       FILL_FIELD(37,5,36,' ');
       BW_PLAN:='Ignore';
       END;
     'S':BEGIN
       FILL_FIELD(25,5,48,' ');
       OTGOXY(25,5); WRITELN('Spray whenever an outbreak occurrs.');
       BW_PLAN:='Spray';
       END;
     'M':SET_BW_MON_PLAN;
     END;                (* of case *)
   IF (CH1='S') OR (CH1='M') THEN BEGIN
     SET_BW_SPRAY_TYPE;
     SET_YRS_BW_PLAN;
     BW_PLAN_WRITEUP;
     END;
   END;                          (* of Set budworm plan *)

PROCEDURE WV_PLAN_WRITEUP;
(*****************************************************************
 * -Called form SET_WV_PLAN                                      *
 * -Records weevil plan on the screen.                           *
 *****************************************************************)
 BEGIN
```

```
    IF (CH1='S') THEN BEGIN
      GOTOXY(15,14);
      WRITELN('Spray every    years.');
      GOTOXY(27,14);
      WRITELN(WEEV_CHECK:2);
      GOTOXY(15,15);
      WRITELN('Starting from year ',START_WV_PLAN:2,', and ending at year ',
           END_WV_PLAN:2);
      GOTOXY(15,16);
      IF (CH2='A') THEN WRITELN('Aerial application')
      ELSE WRITELN('Ground application');
      END
    ELSE BEGIN
      GOTOXY(15,14);
      WRITELN('Monitor every    years.');
      GOTOXY(29,14);
      WRITELN(WEEV_CHECK:2);
      GOTOXY(15,15);
      WRITELN('Starting from year ',START_WV_PLAN:2,', and ending at year ',
           END_WV_PLAN:2);
      GOTOXY(15,16);
      WRITELN('Spray if more than ',WV_LEV:5:1,'% weeviling is found');
      GOTOXY(15,17);
      IF (CH2='A') THEN WRITELN('Aerial application')
      ELSE WRITELN('Ground application');
      END;
    END;          (* of wv plan writeup *)


PROCEDURE SET_WV_PLAN;
(*****************************************************************
 * -Called from SET_AUTO                                         *
 * -Sets up a plan of action for dealing with the white pine     *
 *  weevil.                                                      *
 *****************************************************************)
  BEGIN
    FILL_FIELD(1,1,80,' ');
    WRITE_STRING(1,1,'Which way will you handle weevil?',33);
    FILL_FIELD(24,13,376,' ');
    CH1:=CH_PROMPT(24,13,CONCAT('I)gnore them, S)pray only, or ',
       'M)onitor and spray'),'$',37,1,['I','i','S','s','M','m']);
    CAP(CH1);
    CASE CH1 OF
      'I':BEGIN
        FILL_FIELD(36,13,37,' ');
        WEEV_PLAN:='Ignore';
        END;
      'S':BEGIN
        FILL_FIELD(24,13,56,' ');
        GOTOXY(25,13); WRITELN('Spray only');
        WEEV_PLAN:='Spray';
        INT_PROMPT(15,14,'How many years between sprayings?','$',
             49,14,WEEV_CHECK,2);
        END;
```

```
      'M':BEGIN
        FILL_FIELD(24,13,56,' ');
        WRITELN('Monitor and Spray');
        WEEV_PLAN:='MonSpray';
        INT_PROMPT(15,15,'How many years between samplings?',
              '$',48,15,WEEV_CHECK,2);
        FILL_FIELD(15,15,40,' ');
        REAL_PROMPT(15,15,CONCAT('If weeviling is found ',
             'to be higher than ??.? percent,$then apply',
             ' control measures.$     Enter a value.'),'$',
             70,17,WV_LEV,5,1);
        FILL_FIELD(15,15,240,' ');
        END;
      END;        (* of case *)
    IF (CH1='S') OR (CH1='M') THEN BEGIN
      CH2:=CH_PROMPT(15,14,CONCAT('Choose from below the type of ',
           'spray operation which will be used.$     A)Aerial ',
           'application $     B) Ground application '),'$',75,16,
           ['A','a','B','b']);
      CAP(CH2);
      WV_SPRAY_TYPE:=CH2;
      FILL_FIELD(15,14,65,' ');
      FILL_FIELD(15,15,65,' ');
      FILL_FIELD(15,16,65,' ');
      REPEAT
        INT_PROMPT(15,15,'What year will we start this plan from?'
            ,'$',55,15,START_WV_PLAN,2);
        UNTIL (START_WV_PLAN >= 0) AND (START_WV_PLAN < 20);
      REPEAT
        INT_PROMPT(15,15,'What year will we end this plan?',
           '$',55,15,END_WV_PLAN,2);
        UNTIL(END_WV_PLAN > START_WV_PLAN) AND (END_WV_PLAN <= 20);
      FILL_FIELD(15,15,65,' ');
      WV_PLAN_WRITEUP;
      END;
    END;                  (* of set wv plan *)

PROCEDURE SET_ROTATION;
(*****************************************************************
 * -Called from SET_AUTO                                         *
 * -Determines how many years to grow the stand before cutting   *
 *****************************************************************)
  BEGIN
    REPEAT
      INT_PROMPT(1,1,'At what age will we cut this stand?','$',38,1,
          ROTATION,2);
      IF (ROTATION<=20) OR (ROTATION>70) THEN BEGIN
        GOTOXY(1,1); WRITELN('********** ERROR, TRY AGAIN *************');
        PAUSE(1);
        END;
      UNTIL (ROTATION<=70) AND (ROTATION>20);
    FILL_FIELD(1,1,45,' ');
    GOTOXY(28,19);
    WRITELN(ROTATION);
```

```
  END;


BEGIN                    (* set auto *)
  BW_SPRAY_TYPE:='A';
  BW_PLAN:='Ignore';
  BW_MON_PLAN:='During outbrks';
  BW_CHECK:=1;
  START_BW_PLAN:=1;
  END_BW_PLAN:=70;
  SM_LARV_LEV:=0.0;
  LG_LARV_LEV:=3.0;
  SAMP_LARGE:=TRUE;
  WEEV_PLAN:='Ignore';
  WEEV_CHECK:=1;
  WV_SPRAY_TYPE:='A';
  WV_LEV:=20.0;
  START_WV_PLAN:=3;
  END_WV_PLAN:=20;
  ROTATION:=40;
  PAGE(OUTPUT);
  GOTOXY(5,3);
  WRITELN('JACK PINE BUDWORM AND WHITE PINE WEEVIL MONITORING AND ',
       'CONTROL PLAN');
  GOTOXY(5,4);
  WRITELN('------------------------------------------------------------',
       '---------');
  GOTOXY(10,5);
  WRITELN('B)udworm plan: (Default = ignore budworm)');
  GOTOXY(10,13);
  WRITELN('W)eevil plan: (Default = ignore weevil)');
  GOTOXY(10,19);
  WRITELN('R)otation length:    years');
  GOTOXY(28,19);
  WRITELN(ROTATION:2);
  GOTOXY(1,21);
  WRITELN('G)o on to the next part of the program');
  REPEAT
    CH:=CH_PROMPT(1,1,'Budworm? Weevil? Rotation? or Go on to next part?'
           ,'$',70,1,['b','B','w','W','r','R','g','G']);
    CAP(CH);
    CASE CH OF
      'B':SET_BW_PLAN;
      'W':SET_WV_PLAN;
      'R':SET_ROTATION;
      'G':;
      END;            (* of case *)
    UNTIL (CH='G');
    SCREEN_MASK('$4:MASK1.TEXT',OK);
    WRITE_STRING(25,3,USER,18);
    WRITE_STRING(24,5,RAN_OR_DET,10);
    WRITE_INT(39,7,NUM_ACRES,3);
    WRITE_INT(33,9,SITE_INDEX,2);
    WRITE_REAL(40,11,DISCOUNT,6,2);
```

```
        WRITE_STRING(35,15,WEEV_LEVEL,8);
      END;              (* of set auto *)

BEGIN       (* of auto choice *)
  CH:=CH_PROMPT(0,0,'A)utomatic management  I)nteractive management',
       '$',50,0,['A','a','I','i']);
  CAP(CH);
  CASE CH OF
    'A':BEGIN
      AUTO:=TRUE;
      SET_AUTO;
      END;
    'I':AUTO:=FALSE;
    END; (* CASE *)
  END; (* OF AUTO_CHOICE *)
```

```
(****************** COST.TEXT ***********************)

SEGMENT PROCEDURE COST_CHOICE;
(*********************************************************************
 * -Called from WELCOME                                              *
 * -Sets up a menu which allows the user to change the wages of      *
 *  personell and to change the costs involved in spraying pesticides *
 *********************************************************************)
 VAR
   INT2:INTEGER;
   OK:BOOLEAN;
   TEMPREAL:REAL;

 PROCEDURE COST_MASK;
 (*********************************************************************
  * -Called from COST_CHOICE                                          *
  * -Places the menu and default cost values on the screen            *
  *********************************************************************)
  BEGIN
    SCREEN_MASK('#4:MASK2.TEXT',OK);
    WRITE_REAL(40,4,SCOUT_WAGE,5,2);
    WRITE_REAL(47,6,TECH_WAGE,5,2);
    WRITE_REAL(36,10,SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT],6,2);
    WRITE_REAL(45,10,SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE],6,2);
    WRITE_REAL(36,12,SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT],6,2);
    WRITE_REAL(45,12,SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE],6,2);
    WRITE_REAL(38,14,SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT],6,2);
    WRITE_REAL(47,14,SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE],6,2);
    WRITE_REAL(38,16,SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT],6,2);
    WRITE_REAL(47,16,SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE],6,2);
    WRITE_REAL(36,19,SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,INTERCEPT],6,2);
    WRITE_REAL(45,19,SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE],6,2);
    WRITE_REAL(36,21,SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT],6,2);
    WRITE_REAL(45,21,SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE],6,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,10,TEMPREAL,8,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,12,TEMPREAL,8,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(65,14,TEMPREAL,8,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(65,16,TEMPREAL,8,2);
    TEMPREAL:=SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,INTERCEPT] +
              SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,19,TEMPREAL,8,2);
    TEMPREAL:=SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT] +
              SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,21,TEMPREAL,8,2);
    INT2:=1;
    END;                          (* of cost mask *)
```

```
PROCEDURE BACCOST;
(*****************************************************************
 * -Called from COST_CHOICE                                      *
 * -Allows the user to change the costs involved with applying   *
 *  chemical pesticides aerially to control budworm. Places new  *
 *  values on the screen.                                        *
 *****************************************************************)
  BEGIN
    REAL_PROMPT(0,0,CONCAT('Budworm-aerial-chemical spray cost:',
                '  --.-- +  --.-- per acre'),'$',37,0,
                SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT],5,2);
    WRITE_REAL(36,10,SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT],
                              5,2);
    REAL_PROMPT(0,0,CONCAT('Budworm-aerial-chemical spray cost:',
                '  --.-- +  --.-- per acre'),'$',46,0,
                SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE],5,2);
    WRITE_REAL(45,10,SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE],
                              5,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,10,TEMPREAL,8,2);
    END;       (* BACCOST *)

PROCEDURE BGCCOST;
(*****************************************************************
 * -Called from COST_CHOICE                                      *
 * -Allows user to change the costs associated with controlling  *
 *  budworm when applying chemical pesticides on the ground. Places *
 *  new values on the screen                                     *
 *****************************************************************)
  BEGIN
    REAL_PROMPT(0,0,CONCAT('Budworm-ground-chemical spray cost:',
                '  --.-- +  --.-- per acre'),'$',37,0,
                SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT],5,2);
    WRITE_REAL(36,12,SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT],
                              5,2);
    REAL_PROMPT(0,0,CONCAT('Budworm-ground-chemical spray cost:',
                '  --.-- +  --.-- per acre'),'$',46,0,
                SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE],5,2);
    WRITE_REAL(45,12,SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE],
                              5,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,12,TEMPREAL,8,2);
    END;     (* BGCCOST *)

PROCEDURE BABCOST;
(*****************************************************************
 * -Called from COST_CHOICE                                      *
 * -Allows user to change costs associated with controlling budworm *
 *  by aerial applications of biological pesticides. Places new  *
 *  values on the screen.                                        *
 *****************************************************************)
```

```
  BEGIN
    REAL_PROMPT(0,0,CONCAT('Budworm-aerial-biological spray cost:',
                ' --.-- + --.-- per acre'),'$',39,0,
                SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT],5,2);
    WRITE_REAL(38,14,SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT],
                          5,2);
    REAL_PROMPT(0,0,CONCAT('Budworm-aerial-biological spray cost:',
                ' --.-- + --.-- per acre'),'$',48,0,
                SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE],5,2);
    WRITE_REAL(47,14,SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE],
                          5,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(67,14,TEMPREAL,8,2);
    END;      (* BABCOST *)

PROCEDURE BGBCOST;
(*********************************************************************
 * -Called by COST_CHOICE                                            *
 * -Allows user to change costs associated with controlling budworm  *
 *  by ground applications of a biological pesticide                 *
 *********************************************************************)
  BEGIN
    REAL_PROMPT(0,0,CONCAT('Budworm-ground-biological spray cost:',
                ' --.-- + --.-- per acre'),'$',39,0,
                SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT],5,2);
    WRITE_REAL(38,16,SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT],
                          5,2);
    REAL_PROMPT(0,0,CONCAT('Budworm-ground-biological spray cost:',
                ' --.-- + --.-- per acre'),'$',48,0,
                SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE],5,2);
    WRITE_REAL(47,16,SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE],
                          5,2);
    TEMPREAL:=SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT] +
              SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(65,16,TEMPREAL,8,2);
    END;      (* BGBCOST *)

PROCEDURE WACCOST;
(*********************************************************************
 * -Called by COST_CHOICE                                            *
 * -Allows user to change costs associated with controlling weevil   *
 *  aerially.                                                        *
 *********************************************************************)
  BEGIN
    REAL_PROMPT(0,0,CONCAT('Weevil-aerial-chemical spray cost:',
                ' --.-- + --.-- per acre'),'$',36,0,
                SPRAY_COST[WEEVIL,AERILA,CHEMICAL,INTERCEPT],5,2);
    WRITE_REAL(36,19,SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,INTERCEPT],
                          5,2);
    REAL_PROMPT(0,0,CONCAT('Weevil-aerial-chemical spray cost:',
                ' --.-- + --.-- per acre'),'$',45,0,
                SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE],5,2);
    WRITE_REAL(45,19,SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE],
```

```
                                5,2);
    TEMPREAL:=SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,INTERCEPT] +
              SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,19,TEMPREAL,8,2);
    END;      (* WACCOST *)

PROCEDURE WGCCOST;
(*********************************************************************
 * -Called by COST_CHOICE                                            *
 * -Allows user to change costs associated with controlling weevil   *
 *  by ground sprays                                                 *
 *********************************************************************)
  BEGIN
    REAL_PROMPT(0,0,CONCAT('Weevil-ground-chemical spray cost:',
                '  --.-- +  --.-- per acre'),'$',36,0,
                SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT],5,2);
    WRITE_REAL(36,21,SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT],
                                5,2);
    REAL_PROMPT(0,0,CONCAT('Weevil-ground-chemical spray cost:',
                '  --.-- +  --.-- per acre'),'$',45,0,
                SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE],5,2);
    WRITE_REAL(45,21,SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE],
                                5,2);
    TEMPREAL:=SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT] +
              SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE]*NUM_ACRES;
    WRITE_REAL(63,21,TEMPREAL,8,2);
    END;      (* WGCCOST *)


BEGIN
  COST_MASK;
  REPEAT
    INT_PROMPT(0,0,CONCAT('Which of the items listed below do you ',
               'wish to change?'),'$',55,0,INT2,2);
    CASE INT2 OF
      1:BEGIN
        REAL_PROMPT(0,0,'Wage of pest scout:  --.-- per hour',
                    '$',21,0,SCOUT_WAGE,5,2);
        WRITE_REAL(40,4,SCOUT_WAGE,5,):2
        END;
      2:BEGIN
        REAL_PROMPT(0,0,'Wage of forest technician:  --.-- per hour',
                        '$',28,0,TECH_WAGE,5,2);
        WRITE_REAL(47,6,TECH_WAGE,5,2);
        END;
      3:BACCOST;
      4:BGCCOST;
      5:BABCOST;
      6:BGBCOST;
      7:WACCOST;
      8:WGCCOST;
      9:(* DO NOTHING *);
      10:EXIT(PROGRAM);
      END; (* OF CASE *)
```

```
  UNTIL (INT2=9);
SCREEN_MASK('$4:MASK1.TEXT',OK);
WRITE_STRING(25,3,USER,18);
WRITE_STRING(24,5,RAN_OR_DET,10);
WRITE_INT(39,7,NUM_ACRES,3);
WRITE_INT(33,9,SITE_INDEX,2);
WRITE_REAL(40,11,DISCOUNT,6,2);
WRITE_STRING(35,15,WEEV_LEVEL,8);
END;                    (* of cost choice *)
```

```
(********************* PREP.TEXT ************************)

SEGMENT PROCEDURE PREP_CHOICE;
(*************************************************************************
 * -Sets up a menu and allows the user to change the site preparation    *
 *  and regeneration method used this run. Default values are placed     *
 *  initially on the menu.                                               *
 *************************************************************************)
 VAR
   CH1:CHAR;
   INT2,INT3:INTEGER;
   OK:BOOLEAN;

 PROCEDURE BURNIT;
 (************************************************************************
  * -Allows user to decide whether or not to burn the site. If a burn    *
  *  is chosen, the user can set the cost for this or accept a           *
  *  default value.                                                      *
  ************************************************************************)
  BEGIN
    CH1:=CH_PROMPT(0,0,'Shall we burn? Y)es  N)o','$',27,0,
             ['Y','y','N','n']);
    CAP(CH1);
    IF (CH1='Y') THEN BEGIN
      BURN:=TRUE;
      WRITE_STRING(18,4,'Yes',3);
      REAL_PROMPT(0,0,'The cost to burn will be ---.-- dollars per acre',
                  '$',25,0,BURN_COST,6,2);
      END
    ELSE BEGIN
      BURN:=FALSE;
      WRITE_STRING(18,4,'No',3);
      END;
    END;       (* BURNIT*)

 PROCEDURE HERBIT;
 (************************************************************************
  * -Allows user to apply a herbicide to the stand. If this is done      *
  *  the cost for this activity can be set or a default value can be     *
  *  used.                                                               *
  ************************************************************************)
  BEGIN
    CH1:=CH_PROMPT(0,0,'Apply a herbicide? Y)es N)o','$',31,0,
         ['Y','y','N','n']);
    CAP(CH1);
    IF (CH1='Y') THEN BEGIN
      HERBICIDE:=TRUE;
      WRITE_STRING(23,6,'Yes',3);
      REAL_PROMPT(0,0,CONCAT('The cost to apply an herbicide will be ',
                  '---.-- dollars per acre'),'$',39,0,HERB_COST,6,2);
      END
    ELSE BEGIN
      HERBICIDE:=FALSE;
      WRITE_STRING(23,6,'No',3);
```

```
      END;
    END;      (* HERBIT *)

PROCEDURE LOPIT;
(*********************************************************************
 * -Allows the user to lop and scatter slash on the site. If this    *
 *  is done the user can set the cost or accept a default value.     *
 *********************************************************************)
  BEGIN
    CH1:=CH_PROMPT(0,0,'Lop and scatter? Y)es N)o','$',30,0,
        ['Y','y','N','n']);
    CAP(CH1);
    IF (CH1='Y') THEN BEGIN
      LOP_SCATTER:=TRUE;
      WRITE_STRING(29,8,'Yes',3);
      END
    ELSE BEGIN
      LOP_SCATTER:=FALSE;
      WRITE_STRING(29,8,'No',3);
      END;
    END;      (* LOPIT *)

PROCEDURE SCARIT;
(*********************************************************************
 * -Allows user to scarify the stand. If this is done, the cost can  *
 *  be set or a default value can be accepted.                       *
 *********************************************************************)
  BEGIN
    CH1:=CH_PROMPT(0,0,'Scarify? Y)es N)o','$',24,0,
        ['Y','y','N','n']);
    CAP(CH1);
    IF (CH1='Y') THEN BEGIN
      SCARIFY:=TRUE;
      WRITE_STRING(21,10,'Yes',3);
      REAL_PROMPT(0,0,CONCAT('The cost to scarify will be ---.-- ',
                    'dollars per acre'),'$',28,0,SCAR_COST,6,2);
      END
    ELSE BEGIN
      SCARIFY:=FALSE;
      WRITE_STRING(21,10,'No',3);
      END;
    END;      (* SCARIT *)

PROCEDURE ROLLIT;
(*********************************************************************
 * -Allows user to rollerchop the stand. If this is done, the cost   *
 *  can be set or a default value can be accepted.                   *
 *********************************************************************)
  BEGIN
    CH1:=CH_PROMPT(0,0,'Rollerchop? Y)es N)o','$',24,0,
        ['Y','y','N','n']);
    CAP(CH1);
    IF (CH1='Y') THEN BEGIN
      ROLLERCHOP:=TRUE;
```

```
      WRITE_STRING(24,12,'Yes',3);
      REAL_PROMPT(0,0,CONCAT('The cost to rollerchop will be ---.-- ',
                    'dollars per acre'),'$',31,0,CHOP_COST,6,2);
      END
    ELSE BEGIN
      ROLLERCHOP:=FALSE;
      WRITE_STRING(24,12,'No',3);
      END;
    END;      (* ROLLIT *)

PROCEDURE REGENIT;
(*****************************************************************
 * -Allows user to choose the method of regeneration to use. Costs  *
 *  can be set or default values can be accepted. Method chosen is  *
 *  printed on the screen.                                          *
 *****************************************************************)
 BEGIN
   FILL_FIELD(40,0,40,' ');
   WRITE_STRING(0,0,CONCAT('Choose one regeneration method from ',
                 'the list below.'),51);
   CH1:=CH_PROMPT(20,15,CONCAT('N)atural regeneration$S)eed with',
        ' a corn planter$A)erial seed$M)achine plant$H)and plant')
        ,'$',53,0,['N','n','S','s','A','a','M','m','H','h']);
   CAP(CH1);
   CASE CH1 OF
     'N':PLANT_METHOD:='Natural regeneration';
     'S':BEGIN
       PLANT_METHOD:='Seed with a corn planter';
       REAL_PROMPT(0,0,CONCAT('Using a corn planter costs ---.-- ',
                  'dollars an acre. (Return= OK, else enter cost)')
                    ,'$',27,0,CP_SEED_COST,6,2);
       END;
     'A':BEGIN
       PLANT_METHOD:='Aerial seed';
       REAL_PROMPT(0,0,CONCAT('Aerial seeding costs ---.-- dollars an ',
                    'acre. (Return = OK, else enter cost)')
                    ,'$',21,0,AIR_SEED_COST,6,2);
       END;
     'M':BEGIN
       PLANT_METHOD:='Machine plant';
       REAL_PROMPT(0,0,CONCAT('Machine planting costs ---.-- dollars ',
                    'an acre. (Return = OK, else enter cost)')
                    ,'$',23,0,MAC_PLANT_COST,6,2);
       END;
     'H':BEGIN
       PLANT_METHOD:='Hand plant';
       REAL_PROMPT(0,0,CONCAT('Hand planting costs ---.-- dollars an ',
                    'acre. (Return = OK, else enter cost)')
                    ,'$',20,0,HND_PLANT_COST,6,2);
       END;
     END; (* CASE *)
   WRITE_STRING(36,14,PLANT_METHOD,24);
   FILL_FIELD(20,15,30,' ');
   FILL_FIELD(20,16,30,' ');
```

```
    FILL_FIELD(20,17,30,' ');
    FILL_FIELD(20,18,30,' ');
    FILL_FIELD(20,19,30,' ');
    END;      (* REGENIT *)


BEGIN
  SCREEN_MASK('$4:MASK3.TEXT',OK);
  WRITE_STRING(18,4,'No',3);
  WRITE_STRING(23,6,'No',3);
  WRITE_STRING(29,8,'No',3);
  WRITE_STRING(21,10,'No',3);
  WRITE_STRING(24,12,'No',3);
  WRITE_STRING(36,14,PLANT_METHOD,24);
  REPEAT
    INT2:=2;INT3:=1;
    INT_PROMPT(0,0,CONCAT('Which of the items listed below do you want ',
                  'to change?'),'$',56,0,INT2,1);
    CASE INT2 OF
      1:BURNIT;
      2:HERBIT;
      3:LOPIT;
      4:SCARIT;
      5:ROLLIT;
      6:REGENIT;
      END; (* CASE *)
    UNTIL (INT2=7);
  SCREEN_MASK('$4:MASK1.TEXT',OK);
  WRITE_STRING(25,3,USER,18);
  WRITE_STRING(24,5,RAN_OR_DET,10);
  WRITE_INT(39,7,NUM_ACRES,3);
  WRITE_INT(33,9,SITE_INDEX,2);
  WRITE_REAL(40,11,DISCOUNT,6,2);
  WRITE_STRING(35,15,WEEV_LEVEL,8);
  END; (* OF PREP_CHOICE *)
```

```
(******************** INITBIOL.TEXT ********************)

SEGMENT PROCEDURE INIT_BIOL;

  PROCEDURE UP_COSTS;
  (*****************************************************************
   * -Called from several procedures within INIT_BIOL              *
   * -Updates a file containing a record of all costs incurred.    *
   *  Items recorded in this file include year of transaction,     *
   *  activity performed, present cost, and discounted cost.       *
   *****************************************************************)
    VAR
      D_RATE:REAL;

    BEGIN
      NUM_EXPENDITURES:=NUM_EXPENDITURES + 1;
      D_RATE:=1 + DISCOUNT/100;
      PRESENT_VALUE:=PRICE*QUANTITY;
      VALUE_DISCOUNTED:=PRESENT_VALUE/EXP(COUNTYR*LN(D_RATE));
      COSTS:=COSTS + VALUE_DISCOUNTED;
      WRITELN(COSTSIN,COUNTYR);
      WRITELN(COSTSIN,ACTIVITY);
      WRITELN(COSTSIN,PRESENT_VALUE);
      WRITELN(COSTSIN,VALUE_DISCOUNTED);
      END; (* Increment_Expenditures *)

  PROCEDURE PLANT;

   (*****************************************************************
    * -Called from INIT_BIOL                                        *
    * -Depending on site prep and regeneration methods will set     *
    *  initial number of trees on the stand.                        *
    *****************************************************************)

    VAR
      CH:CHAR;
      TEMPINT:INTEGER;

  PROCEDURE NATURAL;
  (*****************************************************************
   * -Called from PLANT                                            *
   * -Determines initial number of seeds for stands regenerated    *
   *  naturally. Is designed so that 1)If a burn was performed then*
   *  0 - 900 seedlings will come in with 75% probability and 900 - *
   *  1000 seedlings will come in with 25% probability, 2)If a lop  *
   *  and scatter and scarification were performed then 300 - 900   *
   *  seedlings will come in, 3)If a rollerchop was performed then  *
   *  100 - 900 seedlings will come in, 4)If nothing was done then  *
   *  0 - 400 seedlings will come in.                               *
   *****************************************************************)
    BEGIN
      PRICE:=0;
      IF BURN THEN
        IF (RAN_OR_DET = 'Random') THEN BEGIN
```

```
        TEMPINT:=RANDOM_INTEGER(100);
        IF (TEMPINT<75) THEN N_TRS_W_CONTROL:=12.0*TEMPINT
        ELSE N_TRS_W_CONTROLS:=(-26400.0 + 364.0*TEMPINT);
        END
      ELSE N_TRS_W_CONTROL:=600.0
    ELSE IF (LOP_SCATTER AND SCARIFY) THEN
      IF (RAN_OR_DET = 'Random') THEN
        N_TRS_W_CONTROL:=(RANDOM_INTEGER(600) + 300.0)
      ELSE N_TRS_W_CONTROL:=600.0
    ELSE IF (ROLLERCHOP) THEN
      IF (RAN_OR_DET = 'Random') THEN
        N_TRS_W_CONTROL:=(RANDOM_INTEGER(800) + 100.0)
      ELSE N_TRS_W_CONTROL:=500.0
    ELSE IF (RAN_OR_DET = 'Random') THEN
      N_TRS_W_CONTROL:= 1.0 * RANDOM_INTEGER(400)
    ELSE N_TRS_W_CONTROL:=200.0;
    IF (N_TRS_W_CONTROL<600.0) THEN BEGIN
      PAGE(OUTPUT);
      WRITE_LONG_STRING(10,8,CONCAT('     It is now year 2 and a ',
               'recent inspection$of this stand has revealed that ',
              'i nadequate regeneration$has occurred.$',
               '     You will now have to hand plant the stand$',
               'to get it up to full stocking.'),'$');
      PAUSE(6);
      COUNTYR:=2;
                                (* The following equation assumes that 280
                                 seedlings can be planted in an hour. Scout
                                 wage is used because it would be too
                                 expensive to employ forest techs for such
                                 menial labor. *)
      PRICE:=(900.0 - N_TRS_W_CONTROL)/280*SCOUT_WAGE;
      QUANTITY:=1;
      ACTIVITY:='.....HAND PLANT.....';
      UP_COSTS;
      PLANT_METHOD:='Hand plant';
      N_TRS_W_CONTROL:=900.0;
      END;
    IF (N_TRS_W_CONTROL>2000.0) THEN BEGIN
      PAGE(OUTPUT);
      WRITE_LONG_STRING(15,8,CONCAT('     It is now year 2 and a ',
               'recent inspection$of the stand has revealed that ',
               'excessive regeneration$has occurred. ',
               'You now must clean the stand$'),'$');
      COUNTYR:=2;
      PAUSE(6);
      N_TRS_W_CONTROL:=900.0;
      PRICE:=SCAR_COST;
      QUANTITY:=1;
      ACTIVITY:='...CLEAN THE STAND..';
      UP_COSTS;
      END;
    END;                          (* of natural *)

PROCEDURE CORN_PLANTER;
```

```
(*****************************************************************
 * -Called from PLANT                                            *
 * -Determines the initial number of trees if the stand was seeded *
 *  with a corn planter. If a burn, a scarification, a herbicide  *
 *  application, or a rollerchop were performed then 100 - 900    *
 *  seedlings will come in else 0 - 600 seedlings will come in.   *
 *****************************************************************)
  BEGIN
    PRICE:=CP_SEED_COST; QUANTITY:=1;
    ACTIVITY:='.SEED: CORN PLANTER.';
    UP_COSTS;
    IF BURN OR SCARIFY OR HERBICIDE OR ROLLERCHOP THEN
      IF (RAN_OR_DET = 'Random') THEN
        NUM_TREES_W_CONTROL:=RANDOM_INTEGER(800) + 100
      ELSE NUM_TREES_W_CONTROL:=500
    ELSE
      IF (RAN_OR_DET = 'Random') THEN
        NUM_TREES_W_CONTROL:=RANDOM_INTEGER(600)
      ELSE NUM_TREES_W_CONTROL:=300;
    IF (N_TRS_W_CONTROL<600.0) THEN
      REPEAT
        PAGE(OUTPUT);
        CH:=CH_PROMPT(15,8,CONCAT('One year later very few ',
             'seedlings are$visible. Options available now are:',
             '$     S)eed again   P)lant'),
             '$',40,10,['S','s','P','p']);
        CAP(CH);
        IF (CH='S') THEN BEGIN
          COUNTYR:=COUNTYR + 1;
          PRICE:=CP_SEED_COST; QUANTITY:=1;
          ACTIVITY:='.SEED: CORN PLANTER.';
          UP_COSTS;
          F IBURN OR SCARIFY OR HERBICIDE OR ROLLERCHOP THEN
            IF (RAN_OR_DET = 'Random') THEN
              NUM_TREES_W_CONTROL:=RANDOM_INTEGER(800) + 100
            ELSE NUM_TREES_W_CONTROL:=500
          ELSE
            IF (RAN_OR_DET = 'Random') THEN
              NUM_TREES_W_CONTROL:=RANDOM_INTEGER(600)
            ELSE NUM_TREES_W_CONTROL:=300;
          END
        ELSE IF (CH='P') THEN BEGIN
          COUNTYR:=COUNTYR + 1;
          PRICE:=HND_PLANT_COST; QUANTITY:=1;
          ACTIVITY:='.....HAND PLANT.....';
          UP_COSTS;
          N_TRS_W_CONTROL:=900.0;
          PLANT_METHOD:='Hand plant';
          END;
        UNTIL (N_TRS_W_CONTROL>600.0);
    END;      (* CORN_PLANTER *)

PROCEDURE AIRPLANE;
(*****************************************************************
```

```
* -Called from PLANT                                                    *
* -Determines the initial number of trees if the stand was aerially*
*  seeded. If a burn, a scarification, an herbicide application,    *
*  or a rollerchop were performed then 100 - 800 seedlings will    *
*  come in else 0 - 500 seedlings will come in.                     *
**********************************************************************)
 BEGIN
   PRICE:=AIR_SEED_COST; QUANTITY:=1;
   ACTIVITY:='....SEED: AERIAL....';
   UP_COSTS;
   IF BURN OR SCARIFY OR HERBICIDE OR ROLLERCHOP THEN
     IF (RAN_OR_DET = 'Random') THEN
       NUM_TREES_W_CONTROL:=RANDOM_INTEGER(800)
     ELSE NUM_TREES_W_CONTROL:=400
   ELSE
     IF (RAN_OR_DET = 'Random') THEN
       NUM_TREES_W_CONTROL:=RANDOM_INTEGER(500)
     ELSE NUM_TREES_W_CONTROL:=250;
   IF (N_TRS_W_CONTROL<600.0) THEN
     REPEAT
       PAGE(OUTPUT);
       CH:=CH_PROMPT(15,8,CONCAT('One year later very few ',
            'seedlings are$visible. Options available now are:',
            '$     S)eed again   P)lant'),
            '$',40,10,['S','s','P','p']);
       CAP(CH);
       IF (CH='S') THEN BEGIN
         COUNTYR:=COUNTYR + 1;
         PRICE:=CP_SEED_COST; QUANTITY:=1;
         ACTIVITY:='.SEED: CORN PLANTER.';
         UP_COSTS;
         IF BURN OR SCARIFY OR HERBICIDE OR ROLLERCHOP THEN
           IF (RAN_OR_DET = 'Random') THEN
             NUM_TREES_W_CONTROL:=RANDOM_INTEGER(800)
           ELSE NUM_TREES_W_CONTROL:=400
         ELSE
           IF (RAN_OR_DET = 'Random') THEN
             NUM_TREES_W_CONTROL:=RANDOM_INTEGER(500)
           ELSE NUM_TREES_W_CONTROL:=250;
         END
       ELSE IF (CH='P') THEN BEGIN
         COUNTYR:=COUNTYR + 1;
         PRICE:=HND_PLANT_COST; QUANTITY:=1;
         ACTIVITY:='.....HAND PLANT.....';
         UP_COSTS;
         N_TRS_W_CONTROL:=900.0;
         PLANT_METHOD:='Hand plant';
         END;
       UNTIL (N_TRS_W_CONTROL>600.0);
   END;      (* AIRPLANE *)


BEGIN
  QUANTITY:=1;
```

```
    PRICE:=0;
    IF BURN THEN PRICE:=PRICE + BURN_COST*NUM_ACRES;
    IF ROLLERCHOP THEN PRICE:=PRICE + CHOP_COST*NUM_ACRES;
    IF HERBICIDE THEN PRICE:=PRICE + HERB_COST*NUM_ACRES;
    IF SCARIFY THEN PRICE:=PRICE + SCAR_COST*NUM_ACRES;
    IF LOP_SCATTER THEN PRICE:=PRICE + SCOUT_WAGE*NUM_ACRES;
    ACTIVITY:='..SITE PREPARATION..';
    UP_COSTS;
    IF (PLANT_METHOD = 'Natural regeneration') THEN NATURAL
    ELSE IF (PLANT_METHOD = 'Seed with a corn planter') THEN CORN_PLANTER
    ELSE IF (PLANT_METHOD = 'Aerial seed') THEN AIRPLANE
    ELSE IF (PLANT_METHOD = 'Machine plant') THEN BEGIN
      PRICE:=MAC_PLANT_COST; QUANTITY:=1;
      ACTIVITY:='....MACHINE PLANT...';
      UP_COSTS;
      N_TRS_W_CONTROL:=900.0;
      END
    ELSE BEGIN
      PRICE:=HND_PLANT_COST; QUANTITY:=1;
      ACTIVITY:='.....HAND PLANT.....';
      UP_COSTS;
      N_TRS_W_CONTROL:=900.0;
      END;
    IF (PLANT_METHOD='Natural regeneration') OR
      (PLANT_METHOD='Aerial seed') OR
      (PLANT_METHOD='Seed with a corn planter') THEN
      COUNTYR:=COUNTYR + 2;
    TREE_AGE:=2;
    N_TRS_WO_CONTROL:=N_TRS_W_CONTROL;
    PAGE(OUTPUT);
    GOTOXY(10,9);
    WRITELN('     It is now year ',COUNTYR:2,' and an average of ',
          N_TRS_W_CONTROL:7:1);
    GOTOXY(10,10);
    WRITELN(' seedlings per acre are growing on your stand.');
    PAUSE(6);
    END; (* OF PLANT *)


PROCEDURE SETBUDWORM ;
(*********************************************************************
 * -Called from INITBIOL                                             *
 * -Determines if a budworm outbreak is occurring at the start. If   *
 *  not, then year when 1st outbreak will occur is determined.       *
 *********************************************************************)
  VAR
    TEMPINT:INTEGER;
    LOYRS:INTEGER;

  BEGIN
    BW1_W_CONTROL:=0.0;
    BW1_WO_CONTROL:=0.0;
    BW2_W_CONTROL:=0.0;
    BW2_WO_CONTROL:=0.0;
```

```
    BW3_W_CONTROL:=0.0;
    BW3_WO_CONTROL:=0.0;
    IF (RAN_OR_DET = 'Random') THEN LOYRS:=RANDOM_INTEGER(10)
    ELSE LOYRS:=5;
    IF (LOYRS <= 3) THEN BEGIN
      YEAR_OF_OUTBREAK:=LOYRS + 1;
      NEXTBUDWORM:=0;
      LASTBUDWORM:=0;
      END
    ELSE BEGIN
      YEAR_OF_OUTBREAK:=0;
      NEXTBUDWORM:=LOYRS - 3;
      LASTBUDWORM:=7 - NEXTBUDWORM;
      END;
    END; (*OF SETBUDWORM*)


PROCEDURE SETWEEVIL;
(*********************************************************************
 * -Called from INITBIOL                                             *
 * -Determines what the weevil levels will be like for the rest of   *
 *  the run. Depending on what weevil level was chosen by the user   *
 *  will randomly select the final weevil level to which weevil      *
 *  populations will climb.                                          *
 *********************************************************************)
  VAR
    TEMPINT:INTEGER;
    TEMPREAL:REAL;
    CH:CHAR;

  BEGIN
    WPW_W_CONTROL:=0.0;
    WPW_WO_CONTROL:=0.0;
    FIRST_WEEVIL:=0.0;
    IF (WEEV_LEVEL='low') THEN
      IF (RAN_OR_DET = 'Random') THEN LASTWEEV:=RANDOM_INTEGER(5)
      ELSE LAST_WEEV:=2
    ELSE IF (WEEV_LEVEL='moderate') THEN
      IF (RAN_OR_DET = 'Random') THEN LASTWEEV:=RANDOM_INTEGER(5) + 5
      ELSE LAST_WEEV:=7
    ELSE IF (WEEV_LEVEL='high') THEN
      IF (RAN_OR_DET = 'Random') THEN LASTWEEV:=RANDOM_INTEGER(10) + 10
      ELSE LAST_WEEV:=13
    ELSE
      IF (RAN_OR_DET = 'Random') THEN LASTWEEV:=RANDOM_INTEGER(80) + 20
      ELSE LAST_WEEV:=46
    END; (* OF SETWEEVIL *)


PROCEDURE SETCHECK;
(*********************************************************************
 * -Called from INITBIOL                                             *
 * -If not auto, determines when next inspection of stand will be.   *
 *********************************************************************)
```

```
    BEGIN
      IF (NOT AUTO) THEN BEGIN
        PAGE(OUTPUT);
        GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
        GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
        GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
        GOTOXY(0,8);
        WRITELN('              **** Year: ',COUNTYR:2,' ****');
        WRITELN;
        IF (YEAR_OF_OUTBREAK=0) THEN
          WRITELN('     The last budworm outbreak in this region',
                  ' occurred ',LASTBUDWORM:3,' years ago.')
        ELSE
          WRITELN('               A budworm outbreak is occurring ',
                  'this year.');
        WRITELN;
        INT_PROMPT(0,11,CONCAT('     How many years should we wait before ',
            'inspecting the stand?$$                    I''ll take a "0" to ',
             'mean: "Inspect now!!!!"'),'$',70,11,CHECK,3);
      END;
    END;                          (* of set check *)

BEGIN                             (* initbiol *)
  PLANT;
  SET_BUDWORM;
  SET_WEEVIL;
  SET_CHECK;
  END;
```

```
(******************** INIT.TEXT ************************)

PROCEDURE INCREMENT_EXPENDITURES;
(*********************************************************************
 * -Called whenever a new expense is incurred.                       *
 * -Updates file which keeps track of all expenditures. Included in  *
 *  this file is year of transaction, type of transaction, cost, and *
 *  cost discounted to initial stand year.                           *
 *********************************************************************)
  VAR
    D_RATE:REAL;

  BEGIN
    NUM_EXPENDITURES:=NUM_EXPENDITURES + 1;
    D_RATE:=1 + DISCOUNT/100;
    PRESENT_VALUE:=PRICE*QUANTITY;
    VALUE_DISCOUNTED:=PRESENT_VALUE/EXP(COUNTYR*LN(D_RATE));
    COSTS:=COSTS + VALUE_DISCOUNTED;
    WRITELN(COSTSIN,COUNTYR);
    WRITELN(COSTSIN,ACTIVITY);
    WRITELN(COSTSIN,PRESENT_VALUE);
    WRITELN(COSTSIN,VALUE_DISCOUNTED);
    IF IS_IT_CONTROL THEN
      CONTROL_COSTS:=CONTROL_COSTS + VALUE_DISCOUNTED;
    END; (*INCREMENT_EXPENDITURES*)


PROCEDURE INITIALIZE;
(*********************************************************************
 * -Called from the main program BANKSIANA                           *
 * -Initializes parameters of the program and gives user option to    *
 *  change them through the use of menus.                            *
 *********************************************************************)

  PROCEDURE USER_TYPE;
  (*******************************************************************
   * -Called from WELCOME                                            *
   * -Determines the point of view of the user, whether he or she is *
   *  from a public agency, private industry, or a private landowner. *
   *  NOTE: Actually this procedure does not do anything which will  *
   *  affect any of the rest of the program. It will eventually be   *
   *  removed.                                                       *
   *******************************************************************)
    VAR INT2:INTEGER;

    BEGIN
      INT2:=1;
      INT_PROMPT(0,0,CONCAT('Type of user: 1) Public agency',
                     ' 2) Private industry 3) Private land owner'),
                     '$',73,0,INT2,1);
      CASE INT2 OF
        1:USER:='Pbluic agency';
        2:USER:='Private industry';
        3:USER:='Private land owner';
```

```
      END; (* CASE *)
    WRITE_STRING(25,3,USER,18);
    END;                        (* of user type *)

PROCEDURE RUN_TYPE;
(*********************************************************************
 * -Called from WELCOME                                              *
 * -Allows user to have some values obtained from the program be     *
 *  random in nature or to have all values obtained be determined.   *
 *********************************************************************)
  VAR INT2:INTEGER;

  BEGIN
    INT2:=1;
    INT_PROMPT(0,0,'Type of run: 1) Random 2) Determined',
                   '$',37,0,INT2,1);
    CASE INT2 OF
      1:RAN_OR_DET:='Random';
     2: RAN_OR_DET:='Determined';
     END; (* CASE *)
    WRITE_STRING(24,5,RAN_OR_DET,10);
    END;                  (* of run type *)

PROCEDURE ACRE_CHOICE;
(*********************************************************************
 * -Called from WELCOME                                              *
 * -Allows user to set number of acres on the stand.                 *
 *********************************************************************)
  BEGIN
    REPEAT
      INT_PROMPT(0,0,'Number of acres (10 - 999):','$',
                 29,0,NUM_ACRES,3);
      UNTIL (NUM_ACRES>=10) AND (NUM_ACRES<=999);
    WRITE_INT(39,7,NUM_ACRES,3);
    END;                                (* of acre choice *)

PROCEDURE SI_CHOICE;
(*********************************************************************
 * -Called from WELCOME                                              *
 * -Allows user to set the site index of the stand.                  *
 *********************************************************************)
  BEGIN
    REPEAT
      INT_PROMPT(0,0,'Site index (45 - 60):','$',23,0,SITE_INDEX,2);
      UNTIL (SITE_INDEX>=45) AND (SITE_INDEX<=60);
    WRITE_INT(33,9,SITE_INDEX,2);
    END;

PROCEDURE RATE_CHOICE;
(*********************************************************************
 * -Called from WELCOME                                              *
 * -Allows the user to set the discount rate.                        *
 *********************************************************************)
  BEGIN
```

```
    REPEAT
      REAL_PROMPT(0,0,'Discount rate % (0.0 - 15.0):','$',
                  30,0,DISCOUNT,6,2);
      UNTIL (DISCOUNT>=0.0) AND (DISCOUNT<=15.0);
    WRITE_REAL(40,11,DISCOUNT,6,2);
    END; (* OF RATE_CHOICE *)

PROCEDURE WEEV_CHOICE;
(*********************************************************************
 * -Allows user to set initial weevil levels.                        *
 *********************************************************************)
  VAR INT2:INTEGER;

  BEGIN
    INT2:=2;
    INT_PROMPT(0,0,CONCAT('Weevils: 1)Low (0-5%) 2)Moderate',
               ' (6-10%) 3)High (11-20%) 4)Extreme (>20%)'),
               '$',75,0,INT2,1);
    CASE INT2 OF
      1:WEEV_LEVEL:='low';
      2:WEEV_LEVEL:='moderate';
      3:WEEV_LEVEL:='high';
      4:WEEV_LEVEL:='extreme';
      END; (*CASE*)
    WRITE_STRING(35,15,WEEV_LEVEL,8);
    END; (* OF WEEV_CHOICE *)


PROCEDURE WELCOME;
(*********************************************************************
 * -Called from INITIALIZE                                           *
 * -Sets up a menu which allows user to change parameter values.     *
 *********************************************************************)
  VAR
    CH:CHAR;
    OK:BOOLEAN;
    INT1:INTEGER; (* USED IN VARIOUS LEVELS OF THE MENU *)

  BEGIN
    SCREEN_MASK('#4:HELLO.TEXT',OK);
    PAUSE(10);
    PAGE(OUTPUT);
    CH:=CH_PROMPT( 0,9,CONCAT('     You have the option to change the way',
               ' certain parts of this program operate.$',
               'Do you wish to see the menu and make some changes?',
               ' .. Y or N.'),'$',63,10,['Y','y','N','n']);
    CAP(CH);
    IF (CH='Y') THEN BEGIN
      SCREEN_MASK('#4:MASK1.TEXT',OK);
      WRITE_STRING(25,3,USER,18);
      WRITE_STRING(24,5,RAN_OR_DET,10);
      WRITE_INT(39,7,NUM_ACRES,3);
      WRITE_INT(33,9,SITE_INDEX,2);
      WRITE_REAL(40,11,DISCOUNT,6,2);
```

```
      WRITE_STRING(35,15,WEEV_LEVEL,8);
      INT1:=0;
      REPEAT
        INT_PROMPT(0,0,CONCAT('Which of the items listed below do you ',
                   'want to change?'),'$',55,0,INT1,2);
        CASE INT1 OF
          1:USER_TYPE;
          2:RUN_TYPE;
          3:ACRE_CHOICE;
          4:SI_CHOICE;
          5:RATE_CHOICE;
          6AUTO_CHOICE;
          7:WEEV_CHOICE;
          8:COST_CHOICE;
          9:PREP_CHOICE;
          10:(* DO NOTHING *);
          11:EXIT(PROGRAM);
          END; (* CASE *)
        UNTIL (INT1=10);
      END;
    END;

PROCEDURE SET_BUG_VALUES;
(*************************************************************************
 * -Called from INITIALIZE                                               *
 * -Initializes some values used in association with insect impact.      *
 *************************************************************************)
  BEGIN
    BUD_TABLE[1,1]:=1.00;
    BUD_TABLE[1,2]:=1.00;
    BUD_TABLE[1,3]:=1.00;
    BUD_TABLE[2,1]:=0.92;
    BUD_TABLE[2,2]:=0.53;
    BUD_TABLE[2,3]:=0.94;
    BUD_TABLE[3,1]:=0.84;
    BUD_TABLE[3,2]:=0.33;
    BUD_TABLE[3,3]:=0.99;
    BUD_TABLE[4,1]:=0.78;
    BUD_TABLE[4,2]:=0.07;
    BUD_TABLE[4,3]:=0.54;
    BUD_TABLE[5,1]:=0.73;
    BUD_TABLE[5,2]:=0.02;
    BUD_TABLE[5,3]:=0.14;
    HIT_0X_WO_CONTROL:=1.0;
    HIT_1X_WO_CONTROL:=0.0;
    HIT_2X_WO_CONTROL:=0.0;
    HIT_3X_WO_CONTROL:=0.0;
    HIT_4X_WO_CONTROL:=0.0;
    HIT_0X_W_CONTROL:=1.0;
    HIT_1X_W_CONTROL:=0.0;
    HIT_2X_W_CONTROL:=0.0;
    HIT_3X_W_CONTROL:=0.0;
    HIT_4X_W_CONTROL:=0.0;
    END;                    (* of set bug values *)
```

```
PROCEDURE SET_COST_VALUES;
(*********************************************************************
 * -Called from INITIALIZE                                           *
 * -Set default costs.                                               *
 *********************************************************************)
  BEGIN
    SCOUT_WAGE:=5.0;
    TECH_WAGE:=8.0;
    SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT]:=75.0;
    SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE]:=5.0;
    SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT]:=75.0;
    SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE]:=5.0;
    SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT]:=75.0;
    SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE]:=5.0;
    SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT]:=75.0;
    SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE]:=5.0;
    SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,INTERCEPT]:=75.0;
    SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE]:=5.0;
    SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT]:=75.0;
    SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE]:=5.0;
    BURN_COST:=25.00;
    CHOP_COST:=12.00;
    HERB_COST:=50.00;
    CP_SEED_COST:=45.00;
    AIR_SEED_COST:=50.00;
    MAC_PLANT_COST:=80.00;
    HND_PLANT_COST:=65.00;
    SCAR_COST:=70.00;
    END;                        (* of set cost values *)


PROCEDURE SETVALUES;
(*********************************************************************
 * -Called from INITIALIZE                                           *
 * -Set miscellaneous values.                                        *
 *********************************************************************)
  BEGIN
    DISCOUNT:=3.0;
    CHECK:=0;
    USER:='Public agency';
    RAN_OR_DET:='Random';
    WEEV_LEVEL:='moderate';
    PLANT_METHOD:='Machine plant';
    BURN:=FALSE;
    ROLLERCHOP:=FALSE;
    HERBICIDE:=FALSE;
    SCARIFY:=FALSE;
    LOP_SCATTER:=FALSE;
    COUNTYR:=0;
    N_TRS_W_CONTROL:=0;
    N_TRS_WO_CONTROL:=0;
    HEIGHT:=1.0;
    P_LIVE_W_CONTROL:=1.0;
```

```
    P_LIVE_WO_CONTROL:=1.0;
    P_GRO_W_CONTROL:=0.0;
    P_GRO_WO_CONTROL:=0.0;
    NUM_EXPENDITURES:=0;
    FINISHED:=FALSE;
    WEEVSPRAY:=FALSE;
    NUMACRES:=40;
    COSTS:=0.0;
    CONTROL_COSTS:=0.0;
    CUT:=FALSE;
    SITEINDEX:=55;
    SPRAY_BUD:=FALSE;
    SPRAY_WV:=FALSE;
    BUD_COUNT:=0;
    WEEV_COUNT:=0;
    TOGGLE:=TRUE;
    READIN:='$4:ALT1COST.TEXT';
    READOUT:='$4:ALT2COST.TEXT';
    REWRITE(COSTSIN,READIN);
    END;                              (* Of set values *)
BEGIN
  GETSEED;
  SET_BUG_VALUES;
  SET_COST_VALUES;
  SETVALUES;
  WELCOME;
  END;
```

```
(*************** SPRAY.TEXT ************************)

  PROCEDURE BUGSPRAY;
  (*********************************************************************
   * -Called from MANAGE                                               *
   * -Alters the severity of either insect if the user chooses to      *
   *  apply a pesticide.                                               *
   *********************************************************************)
    VAR
      TEMPREAL:REAL;
      TEMPINT:INTEGER;

    PROCEDURE AIR_CHEM_SPRAY;
    (*********************************************************************
     * -Called from SPRAY_BUDWORM                                        *
     * -Alters budworm severity if an aerial application of a chemical   *
     *  pesticide is chosen by the user. 90 - 100% control occurs.       *
     *********************************************************************)
     BEGIN
       IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(10)
       ELSE TEMPINT:=5;
       BW1_W_CONTROL:=(TEMPINT/100)*BW1_W_CONTROL;
       IS_IT_CONTROL:=TRUE;
       PRICE:=SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,INTERCEPT] +
                SPRAY_COST[BUDWORM,AERIAL,CHEMICAL,SLOPE]*NUM_ACRES;
       PRICE:=PRICE/NUM_ACRES;
       QUANTITY:=1;
       ACTIVITY:='CHEM. PEST. BUDWORM.';
       INCREMENT_EXPENDITURES;
       END;                          (* of air chem spray *)

  PROCEDURE AIR_BIO_SPRAY;
  (*********************************************************************
   * -Called from SPRAY_BUDWORM                                        *
   * -Alters budworm severity if an aerial application of a            *
   *  biological pesticide is chosen by the user. 70 - 90% control.    *
   *********************************************************************)
    BEGIN
      IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(20) + 10
      ELSE TEMPINT:=20;
      BW1_W_CONTROL:=(TEMPINT/100)*BW1_W_CONTROL;
      IS_IT_CONTROL:=TRUE;
      PRICE:=SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,INTERCEPT] +
               SPRAY_COST[BUDWORM,AERIAL,BIOLOGICAL,SLOPE]*NUM_ACRES;
      PRICE:=PRICE/NUM_ACRES;
      QUANTITY:=1;
      ACTIVITY:='BIOL. PEST. BUDWORM.';
      INCREMENT_EXPENDITURES;
      END;                               (* of air bio spray *)

  PROCEDURE GRND_CHEM_SPRAY;
  (*********************************************************************
   * -Called from SPRAY_BUDWORM                                        *
   * -Alters budworm severity if a ground application of a chemical    *
```

```
 *  pesticide is chosen by the user. 90 - 100% control occurs.     *
 ****************************************************************)
  BEGIN
    IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(10)
    ELSE TEMPINT:=5;
    BW1_W_CONTROL:=(TEMPINT/100)*BW1_W_CONTROL;
    IS_IT_CONTROL:=TRUE;
    PRICE:=SPRAY_COST[BUDWORM,GROUND,CHEMICAL,INTERCEPT] +
             SPRAY_COST[BUDWORM,GROUND,CHEMICAL,SLOPE]*NUM_ACRES;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='CHEM. PEST. BUDWORM.';
    INCREMENT_EXPENDITURES;
    END;                                  (* of grnd chem spray *)

PROCEDURE GRND_BIO_SPRAY;
(****************************************************************
 * -Called from SPRAY_BUDWORM                                   *
 * -Alters budworm severity if a ground application of a biological*
 *  pesticide is chosen by the user. 70 - 90% control occurs.    *
 ****************************************************************)
  BEGIN
    IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(20) + 10
    ELSE TEMPINT:=20;
    BW1_W_CONTROL:=(TEMPINT/100)*BW1_W_CONTROL;
    IS_IT_CONTROL:=TRUE;
    PRICE:=SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,INTERCEPT] +
             SPRAY_COST[BUDWORM,GROUND,BIOLOGICAL,SLOPE]*NUM_ACRES;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='BIOL. PEST. BUDWORM.';
    INCREMENT_EXPENDITURES;
    END;                              (* of grnd bio spray *)


PROCEDURE SPRAY_BWORM;
(****************************************************************
 * -Called from BUGSPRAY                                        *
 * -Type of pesticide application to be used against the budworm *
 *  is chosen.                                                    *
 ****************************************************************)
  BEGIN
    IF AUTO THEN CH:=BW_SPRAY_TYPE
    ELSE BEGIN
      PAGE(OUTPUT);
      GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
      GOTOXY(50,1);WRITELN('$ Acres     : ',NUM_ACRES:2);
      GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
      GOTOXY(0,10);
      WRITELN('             HOW SHALL WE DO THIS?');
      WRITELN('               A) Aerial application / Chemical pesticide');
      WRITELN('               B) Aerial application / Biological pesticide');
      WRITELN('               C) Ground application / Chemical pesticide');
      WRITELN('               D) Ground application / Biological pesticide');
```

```
      READ(KEYBOARD,CH);
      CAP(CH);
      END;
    CASE CH OF
      'A':AIR_CHEM_SPRAY;
      'B':AIR_BIO_SPRAY;
      'C':GRND_CHEM_SPRAY;
      'D':GRND_BIO_SPRAY;
      END; (* CASE *)
    END; (* SPRAY_BWORM *)

PROCEDURE AIR_SPRAY;
(*****************************************************************
 * -Called from SPRAY_WEEVIL                                     *
 * -Alters weevil severity if an aerial pesticide spray is chosen. *
 *  40 - 50% control 1st year, 15 - 30% control 2nd year.        *
 *****************************************************************)
  BEGIN
    IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(10) + 50
    ELSE TEMPINT:=55;
    WPW_W_CONTROL:=(TEMPINT/100)*WPW_W_CONTROL;
    IS_IT_CONTROL:=TRUE;
    PRICE:=SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,INTERCEPT] +
             SPRAY_COST[WEEVIL,AERIAL,CHEMICAL,SLOPE]*NUM_ACRES;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='.CHEM. PEST. WEEVIL.';
    INCREMENT_EXPENDITURES;
    END;                                    (* of air spray *)

PROCEDURE GRND_SPRAY;
(*****************************************************************
 * -Called from SPRAY_WEEVIL                                     *
 * -Alters weevil severity if a ground application of pesticide is *
 *  chosen. 40 - 50% control.                                    *
 *****************************************************************)
  BEGIN
    IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(10) + 50
    ELSE TEMPINT:=55;
    WPW_W_CONTROL:=(TEMPINT/100)*WPW_W_CONTROL;
    IS_IT_CONTROL:=TRUE;
    PRICE:=SPRAY_COST[WEEVIL,GROUND,CHEMICAL,INTERCEPT] +
             SPRAY_COST[WEEVIL,GROUND,CHEMICAL,SLOPE]*NUM_ACRES;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='.CHEM. PEST. WEEVIL.';
    INCREMENT_EXPENDITURES;
    END;                                    (* of grnd spray *)


PROCEDURE SPRAY_WEEVIL;
(*****************************************************************
 * -Called from BUGSPRAY                                         *
 * -Allows user to choose what type of pesticide application to  *
```

```
*  direct against the weevil.                                          *
**********************************************************************)
 BEGIN
   IF AUTO THEN CH:=WV_SPRAY_TYPE
   ELSE BEGIN
     PAGE(OUTPUT);
     CH:=CH_PROMPT(15,8,CONCAT('HOW SHALL WE DO THIS?$       ',
       'A) Aerial spray$       B) Ground spray'),'$',39,10,['A','a','B','b']
     CAP(CH);
     END;
   CASE CH OF
     'A':AIR_SPRAY;
     'B':GRND_SPRAY;
     END; (* OF CASE *)
   WEEV_SPRAY:=TRUE;
   END; (* OF SPRAY_WEEVIL *)


BEGIN                    (* bugspray *)
  IF AUTO THEN BEGIN
    IF SPRAY_BUD THEN CH:='A';
    IF SPRAY_WV THEN CH:='B';
    IF (CH='A') THEN SPRAY_BUD:=FALSE;
    IF (CH='B') THEN SPRAY_WV:=FALSE;
    END
  ELSE BEGIN
    PAGE(OUTPUT);
    GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
    GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
    GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
    GOTOXY(0,10);
    IF (COUNTYR <= 20) THEN BEGIN
      WRITELN('          TOWARDS WHICH INSECT WILL WE DIRECT THIS SPRAYING?'
      WRITELN('                                  A) JACK PINE BUDWORM');
      WRITELN('                                  B) WHITE PINE WEEVIL');
      WRITELN('                                  C) OOPS, WRONG BUTTEN! ',
                                                  'I DON''T WANT TO SPRAY!');
      EAD(KEYBOARD,CH);
      END
     ELSE CH:='A';
    END;
  CAP(CH);
  CASE CH OF
    'A':SPRAY_BWORM;
    'B':SPRAY_WEEVIL;
    'C':EXIT(BUG_SPRAY);
    END; (* CASE *)
  IF (NOT AUTO) THEN BEGIN
    PAGE(OUTPUT);
    GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
    GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
    GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
    GOTOXY(20,10);
    WRITELN('Spraying Completed.');
```

```
  PAUSE(6);
  EN                                (* of bugspray *)
END; (* OF BUGSPRAY *)
```

```
(* ************** MANAGE.TEXT ************* *)

PROCEDURE MANAGE;
(************************************************************************
 * -Called from MONITOR                                                 *
 * -Allows user to apply pesticides, inventory the stand, get a         *
 *  listing of costs, cut the stand, and/or exit the program.           *
 ************************************************************************)
  VAR
    CH1:CHAR;
    TEMPINT:INTEGER;
    TEMPREAL:REAL;


  PROCEDURE DESCRIBESTAND;
  (**********************************************************************
   * -Called from REPORT                                                *
   * -Determines the proportion of trees rendered unmerchantable by     *
   *  the weevil and the proportion showing some damage but still       *
   *  merchantable.                                                     *
   **********************************************************************)
     VAR
       USELESS:REAL;
       DAMAGED:REAL;

     BEGIN
       USELESS:=HIT_4X_W_CONTROL*100;
       DAMAGED:=(HIT_1X_W_CONTROL + HIT_2X_W_CONTROL +
                          HIT_3X_W_CONTROL)*100;
       WRITELN;
       WRITELN('    - ',USELESS:5:1,' percent of the stand is too damaged by',
               ' weevil to be marketable.');
       WRITELN('    - ',DAMAGED:5:1,' percent of the stand shows obvious',
               ' signs of weevil damage but still ');
       WRITELN('          is marketable.');
       END; (* OF DESCRIBE_STAND *)

 PROCEDURE REPORTSEED;
 (**********************************************************************
  * -Called from REPORT                                                *
  * -Puts on the screen reports of % mortality and proportion          *
  *  damaged trees for stands under 20 years old.                      *
  **********************************************************************)
   VAR
     CH:CHAR;
     TEMPREAL:REAL;

   BEGIN
     TEMPREAL:=(1 - P_LIVE_W_CONTROL)*100.0;
     PAGE(OUTPUT);
     GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
     GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
     GOTOXY(50,2);WRITELN('Site index : ',SITEINDEX:2);
     GOTOXY(0,8);
```

```
    WRITELN('                  ***  Year: ',COUNTYR:2,' ***');
    WRITELN;
    WRITELN('                       At age ',TREE_AGE:2,':');
    WRITELN;
    WRITELN('                - ',TEMPREAL:4:1,' % mortality has occurred.');
    WRITELN;
    DESCRIBE_STAND;
    IS_IT_CONTROL:=FALSE;
    PRICE:=9.00 + TECH_WAGE*NUM_ACRES*0.05;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='CHECK ON YOUNG STAND';
    INCREMENT_EXPENDITURES;
    WRITELN;
    WRITELN('             (RETURN)');
    READLN(CH);
    END; (* OF REPORTSEED *)

PROCEDURE GET_INV_COST;
(*********************************************************************
 * -Called from REPORT_TREE                                          *
 * -Determines the cost incurred if the stand was inventoried.       *
 *  Price used comes from assuming a fixed cost to arrive at the     *
 *  site plus variable cost depending on how many point samples      *
 *  would need to be taken. Num. samples taken from Marty extension  *
 *  bulletin E-1757; 0.25 hours assumed needed at each point.        *
 *********************************************************************)
 BEGIN
   IF (NUM_ACRES <= 40)
     THEN BEGIN
       PRICE:=9.00 + 0.25*NUMACRES*TECH_WAGE;
       PRICE:=PRICE/NUM_ACRES;
       END
   ELSE IF (NUM_ACRES <= 80)
     THEN BEGIN
       PRICE:=9.00 + 0.25*TECH_WAGE*(40 + (NUM_ACRES - 40)/2);
       PRICE:=PRICE/NUM_ACRES;
       END
   ELSE IF (NUM_ACRES <=240)
     THEN BEGIN
       PRICE:=9.00 + 0.25*TECH_WAGE*(60 + (NUM_ACRES - 80)/4);
       PRICE:=PRICE/NUM_ACRES;
       END
   ELSE BEGIN
     PRICE:=9.00 + 0.25*TECH_WAGE*100;
     PRICE:=PRICE/NUM_ACRES;
     END;
   END;      (* GET_INV_COST *)


PROCEDURE REPORTTREE;
(*********************************************************************
 * -Called from REPORT                                               *
 * -Conducts an inventory of the stand. Reports on the volume found,*
```

```
*  amount of that volume which is damaged by weevil, and on the    *
*  price which would be required per cord to break even given the  *
*  total costs so far and total volume harvested.                  *
*******************************************************************)
 VAR
   CH:CHAR;
   J:INTEGER;
   TEMPREAL:REAL;
   COUNT:REAL;
   INTCOUNT:INTEGER;
   PROP_GOOD:REAL;

 BEGIN
   PAGE(OUTPUT);
   PROP_GOOD:=HIT_0X_W_CONTROL + HIT_1X_W_CONTROL + HIT_2X_W_CONTROL +
                    HIT_3X_W_CONTROL;
   GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
   GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
   GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
   GOTOXY(0,7);
   WRITELN('                    RESULTS OF YEAR ',COUNTYR:2,' INVENTORY:');
   WRITELN;
   WRITELN('           - Your stand is ',TREE_AGE:2,' years old.');
   WRITELN('           - ',NUM_ACRES:2,' acres have been planted.');
   WRITELN('           - There are ',N_TRS_W_CONTROL:6:1,' trees per acre.');
   WRITELN('           - Basal area per acre is ',BA_W_CONTROL:5:1,
                                    ' squre feet.');
   WRaITELN('            - The number of cords per acre are ',
       CORDS_W_CO NTROL:5:1);
   TEMPREAL:=HIT_4X_W_CONTROL*100;
   WRITELN('           - ',TEMPREAL:5:1,' % of this has been rendered ',
               'useless by the weevil');
   TEMPREAL:=PROP_GOOD*100;
   WRITELN('           - ',TEMPREAL:5:1,' % of the trees in this stand ',
               'show signs of damage ');
   WRITELN('               but still are marketable');
   GOTOXY(15,18);
   TEMPREAL:=PROP_GOOD*CORDS_W_CONTROL;
   TEMPREAL:=(COSTS*EXP(COUNTYR*LN(1 + DISCOUNT/100)))/TEMPREAL;
   WRITELN('At this volume a price of ',TEMPREAL:6:2,
        ' dollars per cord ');
   GOTOXY(10,19);
   WRITELN('would be required to break even.');
   IS_IT_CONTROL:=FALSE;
   GET_INV_COST;
   QUANTITY:=1;
   ACTIVITY:='...STAND INVENTORY..';
   INCREMENT_EXPENDITURES;
   GOTOXY(0,22);
   WRITELN('                 Press RETURN');
   READLN(CH);
   END; (* OF REPORTTREE*)
```

```
PROCEDURE REPORT;
(*****************************************************************
 * -Called from MANAGE                                           *
 * -Allows user to get pertinant information on the status of the *
 *  stand.  ex. volume, mortality, damage, etc.                  *
 *****************************************************************)
  BEGIN
    IF (TREE_AGE<=20) THEN REPORTSEED
      ELSE REPORTTREE;
    END; (* OF REPORT *)

PROCEDURE CHECK_EXPENDITURES;
(*****************************************************************
 * -Called from MANAGE                                           *
 * -Produces a listing of all costs incurred so far. Included in the *
 *  listing is the year the cost was incurred, the nature of the *
 *  activity for which the cost was incurred, the cost itself, and *
 *  the value of the cost discounted back to the initial year.   *
 *****************************************************************)
 VAR
   I:INTEGER;
   VALUE:REAL;

 BEGIN
   IF TOGGLE THEN BEGIN
     TOGGLE:=FALSE;
     READIN:='$4:ALT2COST.TEXT';
     READOUT:='$4:ALT1COST.TEXT';
     END
   ELSE BEGIN
     TOGGLE:=TRUE;
     READIN:='$4:ALT1COST.TEXT';
     READOUT:='$4:ALT2COST.TEXT';
     END;
   CLOSE(COSTSIN,LOCK);
   REWRITE(COSTSIN,READIN);
   RESET(COSTSOUT,READOUT);
   VALUE:=0.0;
   PAGE(OUTPUT);
   WRITELN('         AGE           ACTIVITY              COST     '
                                                 ,'COST DISCOUNTED');
   WRITELN('-------------------------------------------------------',
            '----------------');
   FOR I:=1 TO NUM_EXPENDITURES DO BEGIN
     READLN(COSTSOUT,YEAR_OF_TRANSACTION);
     WRITELN(COSTSIN,YEAR_OF_TRANSACTION);
     READLN(COSTSOUT,ACTIVITY);
     WRITELN(COSTSIN,ACTIVITY);
     READLN(COSTSOUT,PRESENT_VALUE);
     WRITELN(COSTSIN,PRESENT_VALUE);
     READLN(COSTSOUT,VALUE_DISCOUNTED);
     WRITELN(COSTSIN,VALUE_DISCOUNTED);
     WRITELN(YEAR_OF_TRANSACTION:12,'     ',ACTIVITY,PRESENT_VALUE:14:2,
                                        VALUE_DISCOUNTED:14:2);
```

```
      VALUE:=VALUE + VALUE_DISCOUNTED;
      IF (I MOD 20 = 0) THEN BEGIN
        WRITELN;
        WRITELN('Press RETURN to see a continuation of this listing.');
        READLN(CH);
        PAGE(OUTPUT);
        WRITELN('         AGE            ACTIVITY                COST     '
                                                  ,'COST DISCOUNTED');
        WRITELN('-------------------------------------------------------',
                                '---------------------');
        END;
      END;
    WRITELN;
    WRITE('The sum of all costs discounted to the planting year: ');
    WRITELN(VALUE:9:2);
    WRITELN;
    WRITELN('                RETURN');
    READLN(CH);
    CLOSE(COSTSOUT,LOCK);
    END; (* CHECK_EXPENDITURES *)


PROCEDURE CUTEVALUATION;
(*********************************************************************
 * -Called from MANAGE                                               *
 * -Cuts the stand. Gives net present value for the stand at         *
 *  planting and gives the net present value of monitor and control  *
 *  activities at planting.                                          *
 *********************************************************************)
 VAR
   CH1:CHAR;
   I:INTEGER;
   J:INTEGER;
   PRICEPULP:REAL; (* PRICE THAT CAN BE GOTTEN PER CUBIC FOOT *)
   TEMPREAL:REAL;

 PROCEDURE GETPRICE;
 (*********************************************************************
  * -Called from CUT_EVALUATION                                       *
  * -Allows the user to set the price per cord.                       *
  *********************************************************************)
   BEGIN
     PAGE(OUTPUT);
     PRICE_PULP:=7.0;
     REPEAT
       REAL_PROMPT(10,8,CONCAT('What price per cord do you want to use?$$',
             '     (Values usually range from 3.00 to 15.00)'),'$',55,8,
           PRICE_PULP,5,2);
       UNTIL (PRICE_PULP >= 0.00);
     END; (* OF GET_PRICE *)


 BEGIN IF (TREE_AGE>20) THEN BEGIN
     GETPRICE;
```

```
    PAGE(OUTPUT);
    GOTOXY(25,2);
    WRITELN('EVALUATION OF ALL ACTIVITIES');
    GOTOXY(25,3);
    WRITELN('---------------------------');
    GOTOXY(33,4);
    TEMPREAL:=(1 - HIT_4X_W_CONTROL)*CORDS_W_CONTROL;
    WRITELN('The number of cords cut= ',TEMPREAL:6:1);
    GOTOXY(24,5);
    BID:=PRICE_PULP*TEMPREAL;
    TEMPREAL:=BID/EXP(COUNTYR*LN(1 + DISCOUNT/100));
    WRITELN('Discounted value for these cords= ',TEMPREAL:7:2);
    GOTOXY(22,6);
    WRITELN('Discounted value of costs incurred= ',COSTS:7:2);
    TEMPREAL:=TEMPREAL - COSTS;
    GOTOXY(14,7);
    WRITELN('Net present value of the stand at planting= ',TEMPREAL:7:2);
    TEMPREAL:=(1 - HIT_4X_W_CONTROL)*CORDS_W_CONTROL;
    TEMPREAL:=(EXP(COUNTYR*LN(1 + DISCOUNT/100))*COSTS)/TEMPREAL;
    GOTOXY(0,9);
    WRITELN('Price of jack pine which would be required to break even= ',
         TEMPREAL:6:2);
    GOTOXY(15,12);
    WRITELN('EVALUATION OF MONITOR AND CONTROL ACTIVITIES');
    GOTOXY(15,13);
    WRITELN('-------------------------------------------');
    GOTOXY(4,14);
    TEMPREAL:=(1 - HIT_4X_W_CONTROL)*CORDS_W_CONTROL -
                  (1 - HIT_4X_WO_CONTROL)*CORDS_WO_CONTROL;
    WRITELN('Number of cords saved by monitor and control program= ',
         TEMPREAL:6:1);
    TEMPREAL:=(PRICE_PULP*TEMPREAL)/EXP(COUNTYR*LN(1 + DISCOUNT/100));
    GOTOXY(25,15);
    WRITELN('Discounted value of these cords= ',TEMPREAL:7:2);
    GOTOXY(12,16);
    WRITELN('Discounted cost of monitor and spray program= ',
         CONTROL_COST:7:2);
    TEMPREAL:=TEMPREAL - CONTROL_COST;
    GOTOXY(10,17);
    WRITELN('Net present value of monitor and spray program= ',
         TEMPREAL:7:2);
    IF AUTO THEN BEGIN
      GOTOXY(60,22);
      WRITELN('RETURN');
      READLN(CH);
      CHECK_EXPENDITURES;
      END;
    FINISHED:=TRUE;
    CUT:=TRUE;
    END
  ELSE BEGIN
    PAGE(OUTPUT);
    GOTOXY(15,10);
    WRITELN('Your stand is too small to cut.');
```

```
          GOTOXY(0,20);
          WRITELN('            RETURN');
          READLN(CH);
          END;
        END;                    (* of cut evaluation *)

BEGIN                          (* manage *)
  REPEAT
    IF (COUNTYR>=70) THEN CH1:='C'
    ELSE BEGIN
      IF AUTO THEN BEGIN
        IF CUT THEN CH1:='C'
        ELSE IF SPRAY_BUD OR SPRAY_WV THEN CH1:='A'
        ELSE CH1:='E';
        END
      ELSE BEGIN
        PAGE(OUTPUT);
        GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
        GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
        GOTOXY(50,2);WRITELN('Site index : ',SITEINDEX:2);
        GOTOXY(0,8);
        WRITELN('              ****** MANAGEMENT OPTIONS ******');
        WRITELN('                  A) ... SPRAY A PESTICIDE');
        WRITELN('                  B) ... INVENTORY');
        WRITELN('                  C) ... CUT THE STAND');
        WRITELN('                  D) ... CHECK ON EXPENDITURES TO DATE');
        WRITELN('                  E) ... DO NOTHING - GO ON TO NEXT YEAR');
        WRITELN('                  F) ... TERMINATE PROGRAM');
        READ(KEYBOARD,CH1);
        CAP(CH1);
        END;
      END;
    CASE CH1 OF
      'A':BUG_SPRAY;
      'B':REPORT;
      'C':CUT_EVALUATION;
      'D':CHECK_EXPENDITURES;
      'E':;
      'F':BEGIN
        WRITELN;WRITELN;
        WRITELN('                    DO YOU REALLY WANT TO LEAVE US?');
        READ(KEYBOARD,CH1);
        IF (CH1='Y') OR (CH1='y') THEN BEGIN
          WRITELN('                                          BYE-BYE!');
          EXIT(PROGRAM);
          END;
        END;
      END   (* OF CASE *)
    UNTIL (CH1='E') OR (CH1='e') OR (CUT);
END;                                   (* of manage *)
```

```
(********************* MONITOR.TEXT ************************)

  PROCEDURE CHECKSTAND;
  (*****************************************************************
   * -Called from main program BANKSIANA                           *
   * -Allows the user to come in and sample for the insects and/or to  *
   *  perform one of the management options on the stand.           *
   *****************************************************************)

    PROCEDURE SETCHECK;
    (***************************************************************
     * -Called from CHECK_STAND                                    *
     * -If not on automatic run, the user must choose how many years  *
     *  to wait before checking the stand again or performing some  *
     *  management.                                                *
     ***************************************************************)
     BEGIN
       IF (NOT AUTO) THEN BEGIN
         PAGE(OUTPUT);
         GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
         GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
         GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
         GOTOXY(0,9);
         WRITELN('                      *** Year ',COUNTYR:2,'***');
         IF (YEAR_OF_OUTBREAK=0) THEN
           WRITELN('            The last budworm outbreak in this region ',
                   'occurred ',LASTBUDWORM:3,' Years ago.')
         ELSE
           WRITELN('         A budworm outbreak is presently occurring in ',
                   'this region.');
         WRITELN;
         INT_PROMPT(0,12,CONCAT('How many years should we wait before',
                    ' inspecting the stand?'),'$',60,12,CHECK,2);
         EN;
       END; (* OF SETCHECK *)

    PROCEDURE SAMPLE;
    (***************************************************************
     * -Called from CHECK_STAND                                    *
     * -Samples the stand to estimate insect population levels.    *
     ***************************************************************)
     VAR
       CH:CHAR;
       LEAVE:BOOLEAN;

     PROCEDURE BWORMSAMPLE;
     (*************************************************************
      * -Called from SAMPLE                                       *
      * -Samples the stand for budworm.                           *
      *************************************************************)
      VAR
        TEMPREAL:REAL;

      PROCEDURE EARLY_SAMPLE;
```

```
(*************************************************************
 * -Called from BWORMSAMPLE                                  *
 * -Performs an overwintering larval sample. Procedure       *
 *  creates sample results of 0-6.8 larvae per 18 inch branch*
 *  depending on what the actual population level is and on  *
 *  what random error is applied to the population.          *
 *************************************************************)
  BEGIN
    IF (NOT AUTO) THEN BEGIN
      GOTOXY(15,8);
      WRITELN('YEAR ',COUNTYR:2,' OVER WINTERING LARVAL SURVEY');
      GOTOXY(15,9);
      WRITELN('-----------------------------------');
      WRITELN('     Laboratory proceedures are used to extract larvae',
              ' from');
      WRITELN('18 inchbranch samples. This technique is used to get ',
           'a preliminary ');
      WRITELN('idea as to whether or not there are any budworm in the ',
              'stand.');
      WRITELN;
      END;
    IF (RAN_OR_DET = 'Random') THEN TEMPREAL:=RANDOM_INTEGER(15) - 7
    ELSE TEMPREAL:=0.0;
    TEMPREAL:=BW1_W_CONTROL + TEMPREAL;
    IF (TEMPREAL>=3) THEN TEMPREAL:=TEMPREAL*0.97 - 2.9
    ELSE TEMPREAL:=0.0;
    IF (NOT AUTO) THEN
      WRITELN('     ',TEMPREAL:3:1,' Larvae per 18 inch branch were ',
              'found this sampling.');
    IS_IT_CONTRL:=TRUE;
    PRICE:=10.00 + 0.55*SCOUT_WAGE*NUM_ACRES;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='EARLY SAMP. BUDWORM.';
    INCREMENT_EXPENDITURES;
    END; (*EARLY SAMPLE*)

PROCEDURE LATE_SAMPLE;
(*************************************************************
 * -Called from BWORMSAMPLE                                  *
 * -Performs a late larval sample for budworm. Procedure     *
 *  creates sample results of 0-2, 2-3, and 3-10 larvae per  *
 *  branch for low, moderate and high populations of budworm *
 *************************************************************)
  BEGIN
    IF (NOT AUTO) THEN BEGIN
      PAGE(OUTPUT);
      GOTOXY(50,0);WRITELN('Year       : ',COUNTYR:2);
      GOTOXY(50,1);WRITELN('# Acres    : ',NUM_ACRES:2);
      GOTOXY(50,2);WRITELN('Site index : ',SITEINDEX:2);
      GOTOXY(15,8);
      WRITELN('YEAR ',COUNTYR:2,' LATE LARVAL SAMPLING');
      WRITELN('      ----------------------------------------------');
      WRITELN('     15 inch branch samples are collected and checked ',
```

```
                        'for budworm.');
      WRITELN('          0 - 2 Larvae per branch indicates low ',
                        'populations');
      WRITELN('          2 - 3 Larvae per branch indicates moderate',
                        ' populations');
      WRITELN('          More than 3 larvae per branch indicates ',
                        'high populations');
      WRITELN;
      END;
    IF (RAN_OR_DET = 'Random') THEN TEMPREAL:=RANDOM_INTEGER(3) - 1
    ELSE TEMPREAL:=0;
    TEMPREAL:=BW1_W_CONTROL + TEMPREAL;
    IF (TEMPREAL>7) THEN TEMPREAL:=2.3*TEMPREAL - 13.3
    ELSE IF (TEMPREAL>3) THEN TEMPREAL:=0.25*TEMPREAL + 1.25
    ELSE TEMPREAL:=0.6*TEMPREAL;
    IF (NOT AUTO) THEN
      WRITELN('     ',TEMPREAL:4:1,' Larvae per 15 inch branch were ',
          'found this sampling.');
    IS_IT_CONTROL:=TRUE;
    PRICE:=10.00 + 0.25*SCOUT_WAGE*NUM_ACRES;
    PRICE:=PRICE/NUM_ACRES;
    QUANTITY:=1;
    ACTIVITY:='.LATE SAMP. BUDWORM.';
    INCREMENT_EXPENDITURES;
    IF (NOT AUTO) THEN BEGIN
      WRITELN;WRITELN;
      WRITELN('               Press RETURN to continue');
      READLN(CH);
      END;
    END;                        (* of late sample *)

BEGIN
  IF (NOT AUTO) THEN BEGIN
    PAGE(OUTPUT);
    GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
    GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
    GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
    END;
  IF (BW1_W_CONTROL > 0.1) THEN BEGIN
    EARLYSAMPLE;
    IF AUTO THEN BEGIN
      IF (SAMP_LARGE) AND (TEMPREAL>=SM_LARV_LEV) THEN CH:='Y'
      ELSE CH:='N';
      IF (NOT SAMP_LARGE) AND (TEMPREAL>=SM_LARV_LEV) THEN
           SPRAY_BUD:=TRUE;
      END
    ELSE BEGIN
      WRITELN;
      WRITELN('        Should we conduct a late larval survey?');
      READ(KEYBOARD,CH);
      END;
    IF (CH='Y') OR (CH='y') THEN BEGIN
      LATESAMPLE;
      IF (AUTO) AND (TEMPREAL>=LG_LARV_LEV) THEN SPRAY_BUD:=TRUE;
```

```
          END;
        END
      ELSE BEGIN
        IF (NOT AUTO) THEN BEGIN
          GOTOXY(15,9);
          WRITELN('No budworms were found in any sample.');
          GOTOXY(15,20);
          WRITELN('Press RETURN to continue');
          READLN(CH);
          END;
        END;
      END;                          (* of budwormsample *)

PROCEDURE WEEVSAMPLE;
(*****************************************************************
 * -Called from SAMPLE                                           *
 * -Samples the stand for weevil. Adds a random error to the     *
 *  sample. Provides a chart which relates sample results to     *
 *  potential volume losses.                                     *
 *****************************************************************)
 VAR
   TEMPREAL:REAL;
   OK:BOOLEAN;
   TEMPINT:INTEGER;
   ROW,COLUMN:INTEGER;

 BEGIN
   IF (TREE_AGE<=3) THEN BEGIN
     TEMPREAL:=0.0;
     IF (NOT AUTO) THEN BEGIN
       PAGE(OUTPUT);
       GOTOXY(0,10);
       WRITELN('     The stand is too young to have any appreciable ',
               'weeviling');
       PAUSE(6);
       END;
     END
   ELSE BEGIN
     IF (NOT AUTO) THEN BEGIN
       SCREEN_MASK('$4:WEEVCHART.TEXT',OK);
       GOTOXY(25,0);
       WRITELN('Year ',COUNTYR:2,' WEEVIL SAMPLING ');
       WRITELN;
       END;
     IF (RAN_OR_DET = 'Random') THEN
       TEMPREAL:=WPW_W_CONTROL + RANDOM_INTEGER(21) - 10
     ELSE TEMPREAL:=WPW_W_CONTROL;
     IF (TEMPREAL<0.0) THEN TEMPREAL:=0.0;
     IF (TEMPREAL>100.0) THEN TEMPREAL:=100.0;
     IF AUTO THEN
       IF (TEMPREAL>=WV_LEV) THEN SPRAY_WV:=TRUE;
     IF (NOT AUTO) THEN BEGIN
       GOTOXY(10,20);
       WRITELN(TEMPREAL:14:1,' Percent weeviling was found in this ',
```

```
                              'sampling.');
        WRITELN('     Expected damage from this much weevilling is marked ',
                        ' "*" in the chart above.');
        TEMPINT:=TRUNC((SITE_INDEX + 2.5)/5.0);
        COLUMN:=15*TEMPINT - 112;
        TEMPINT:=TRUNC((TEMPREAL + 2.5)/ 5.0);
        IF (TEMPINT =10) AND (TEMPINT<>0) THEN ROW :=TEMPINT + 7
        ELSE IF (TEMPINT=0) THEN ROW:=6
        ELSE BEGIN
          ROW :=18;
          COLUMN:=26;
          END;
        GOTOXY(COLUMN,ROW);
        WRITELN('**');
        IF (ROW=18) THEN COLUMN:=59;
        GOTOXY(COLUMN + 6,ROW);
        WRITELN('**');
        END;
      IS_IT_CONTROL:=TRUE;
      PRICE:=10.00 + 0.05*SCOUT_WAGE*NUM_ACRES;
      PRICE:=PRICE/NUM_ACRES;
      QUANTITY:=1;
      ACTIVITY:='....WEEVIL SAMPLE...';
      INCREMENT_EXPENDITURES;
      IF (NOT AUTO) THEN BEGIN
        GOTOXY(65,22);
        WRITELN('Press RETURN');
        READLN(CH);
        END;
      END;
    END;                          (* of weevsample *)

BEGIN                           (*Sample*)
  IF AUTO THEN
    IF (BW_PLAN = 'MonSpray') AND (WEEV_PLAN = 'MonSpray')
      THEN BEGIN
      IF (NEXT_BUDWORM = 0) AND (COUNTYR >= START_BW_PLAN) AND
           (COUNTYR <= END_BW_PLAN)
        THEN BWORMSAMPLE;
      IF (WEEVCHECK = WEEVCOUNT) AND (COUNTYR >= START_WV_PLAN)
           AND (COUNTYR <= END_WV_PLAN)
        THEN BEGIN
          WEEVSAMPLE;
          WEEV_COUNT:=0;
          END;
      END
    ELSE IF (BW_PLAN = 'MonSpray') AND (COUNTYR >= START_BW_PLAN)
         AND (COUNTYR <= END_BW_PLAN) AND (NEXT_BUDWORM = 0)
      THEN BEGIN
      BWORMSAMPLE;
      END
    ELSE IF (WEEV_PLAN = 'MonSpray') AND (COUNTYR >= START_WV_PLAN)
         AND (COUNTYR <= END_WV_PLAN) AND (WEEV_CHECK = WEEV_COUNT)
      THEN BEGIN
```

```
        WEEVSAMPLE;
        WEEV_COUNT:=0;
        END;
    IF (NOT AUTO) THEN BEGIN
      REPEAT
        PAGE(OUTPUT);
        GOTOXY(50,0);WRITELN('Year        : ',COUNTYR:2);
        GOTOXY(50,1);WRITELN('# Acres     : ',NUM_ACRES:2);
        GOTOXY(50,2);WRITELN('Site index  : ',SITEINDEX:2);
        GOTOXY(14,9);
        WRITELN('*** Year ',COUNTYR:2,'   S A M P L I N G ***');
        IF (COUNTYR<20) THEN BEGIN
          WRITELN('                FOR WHICH INSECT WILL WE SAMPLE?');
          WRITELN('                        A) NONE - GO ON TO MANAGEMENT ',
               'OPTIONS');
          WRITELN('                        B) JACK PINE BUDWORM');
          WRITELN('                        C) WHITE PINE WEEVIL');
          READ(KEYBOARD,CH);
          END
        ELSE BEGIN
          WRITELN('                     SHALL WE SAMPLE FOR BUDWORM?');
          READ(KEYBOARD,CH);
          IF (CH='Y') OR (CH='y') THEN CH:='B'
          ELSE CH:='A';
          END;
        CAP(CH);
        CASE CH OF
          'A':EXIT(SAMPLE);
          'B':BWORMSAMPLE;
          'C':WEEVSAMPLE;
          END; (*CASE*)
        UNTIL (CH = 'A');
      END;
    END;                        (*Sample*)

BEGIN                           (*Checkstand*)
  IF (AUTO) AND (NOT CUT) THEN BEGIN
    IF (BW_PLAN = 'MonSpray') OR (WEEV_PLAN = 'MonSpray') THEN
            SAMPLE;
    IF (BW_PLAN = 'Spray') AND (NEXT_BUDWORM = 0) AND
       (COUNTYR >= START_BW_PLAN) AND (COUNTYR <= END_BW_PLAN)
      THEN SPRAY_BUD:=TRUE;
    IF (WEEV_PLAN = 'Spray') AND (WEEV_COUNT = WEEV_CHECK)
       AND (COUNTYR >= START_WV_PLAN) AND (COUNTYR <= END_WV_PLAN)
      THEN BEGIN
      SPRAY_WV:=TRUE;
      WEEV_COUNT:=0;
      END;
    MANAGE;
    END;
  IF AUTO AND CUT THEN MANAGE;
  IF (NOT AUTO) THEN BEGIN
    SAMPLE;
    MANAGE;
```

```
  IF (CUT) THEN WRITELN
  ELSE SETCHECK;
  END;
END;                                (*Checkstand*)
```

```
(****************** SETPEST.TEXT ***********************)

PROCEDURE SETPESTS;
(*************************************************************************
 * -Called from main program BANKSIANA                                   *
 * -Sets the potential population levels for the budworm and the         *
 *  weevil.                                                              *
 *************************************************************************)
 VAR
   SEED:INTEGER;

 PROCEDURE BUDWORM;
 (************************************************************************
  * -Called from SETPESTS                                                *
  * -Determines if an outbreak year, if it is and user was not           *
  *  planning on inspecting the site, gives user a warning. Sets         *
  *  potlential budworm population levels for the year.                  *
  ************************************************************************)
  VAR
    CH:CHAR;
    TEMPREAL,TEMP:REAL;
    TEMPINT:INTEGER;
    I:INTEGER;

  FUNCTION KEYPRESS: BOOLEAN;
  (************************************************************************
   * -Called from BUDWORM                                                 *
   * -Returns TRUE if any key has been pressed;  Unlike the Apple         *
   *  keypress this function clears the input buffer.                     *
   ************************************************************************)
   VAR
     STATUS: ARRAY[1..30] OF INTEGER;

   BEGIN
     KEYPRESS := FALSE;
     UNITSTATUS( 2, STATUS, 1 );
     IF STATUS[ 1 ] > 0 THEN BEGIN
       KEYPRESS := TRUE;
       UNITCLEAR( 2 );
       END;
     END;                          (* of keypress *)

  PROCEDURE SET_BW_SEVERITY;
  (************************************************************************
   * -Called from BUDWORM                                                 *
   * -For 4 outbreak years, randomly sets population levels so that       *
   *  very low - moderate during first year, moderate - high during       *
   *  second year, moderate - very high during third year,                *
   *  very low - very high during last year. Population levels:           *
   *  0 - 3 = very low, 4 - 5 =low, 6 - 7 = moderate, 8 = high,           *
   *  and 9 = very high.                                                  *
   ************************************************************************)
   BEGIN
     YEAR_OF_OUTBREAK:=YEAR_OF_OUTBREAK + 1;
```

```
    IF (YEAR_OF_OUTBREAK>4) THEN YEAR_OF_OUTBREAK:=4;
    IF (YEAR_OF_OUTBREAK<1) THEN YEAR_OF_OUTBREAK:=1;
    CASE YEAR_OF_OUTBREAK OF
      1:BEGIN
        IF (RAN_OR_DET = 'Random') THEN BW1_W_CONTROL:=RANDOM_INTEGER(8)
        ELSE BW1_W_CONTROL:=3;
        BW1_WO_CONTROL:=BW1_W_CONTROL;
        END;
      2:BEGIN
        IF (RAN_OR_DET = 'Random') THEN
           BW1_W_ CONTROL:=RANDOM_INTEGER(3) + 6
        ELSE BW1_W_CONTROL:=7;
        BW1_WO_CONTROL:=BW1_W_CONTROL;
        END;
      3:BEGIN
        IF (RAN_OR_DET = 'Random') THEN
          BW1_W_CONTROL:=RANDOM_INTEGER(4) + 6
        ELSE BW1_W_CONTROL:=7;
        BW1_WO_CONTROL:=BW1_W_CONTROL;
        END;
      4:BEGIN
        IF (RAN_OR_DET = 'Random') THEN BW1_W_CONTROL:=RANDOM_INTEGER(10)
        ELSE BW1_W_CONTROL:=5;
        BW1_WO_CONTROL:=BW1_W_CONTROL;
        EAYR_OF_OUTBREAK:=0;
        IF (RAN_OR_DET = 'Random') THEN BEGIN
         TE MP:=RANDOM_INTEGER(100)/100;
          IF (TEMP <= 0.0001) THEN TEMP:=0.0001;
          NEXTBUDWORM:=ROUND(-7*LN(TEMP));
          IF (NEXT_BUDWORM>12) THEN NEXT_BUDWORM:=12;
          IF (NEXT_BUDWORM<3) THEN NEXT_BUDWORM:=3;
          END
        ELSE NEXT_BUDWORM:=6;
        LASTBUDWORM:=1;
        END;
      END;(* OF CASE *)
    END;                    (* of set bw severity *)


BEGIN
  BW3_WO_CONTROL:=BW2_WO_CONTROL;
  BW3_W_CONTROL:=BW2_W_CONTROL;
  BW2_W_CONTROL:=BW1_W_CONTROL;
  BW2_WO_CONTROL:=BW1_WO_CONTROL;
  IF (NEXTBUDWORM=0) THEN BEGIN
    IF (NOT AUTO) AND (CHECK<>0) THEN BEGIN
      BEEP;
      PAGE(OUTPUT);
      GOTOXY(10,9);
      WRITELN('*** Year ',COUNTYR:2,' ***');
      WRITELN('              WARNING! Budworm outbreak is now occurring.');
      I:=0;
      WRITELN('        You have 5 seconds to act on this. Press any key ');
      WRITELN('        to halt program.');
```

```
      WHILE ( NOT KEYPRESS ) AND ( I<2000 ) DO BEGIN
        I:=I+1;
        END;
      IF (I<2000) THEN CHECK:=0;
      END
    ELSE IF (NOT AUTO) THEN BEGIN
      BEEP;
      PAGE(OUTPUT);
      GOTOXY(10,9);
      WRITELN('*** Year ',COUNTYR:2,' ***');
      WRITELN('            WARNING! Budworm outbreak is now occurring.');
      PAUSE(3);
      END;
    SET_BW_SEVERITY;
    END
  ELSE BEGIN
    BW1_W_CONTROL:=0;
    BW1_WO_CONTROL:=0;
    NEXTBUDWORM:=NEXTBUDWORM - 1;
    LASTBUDWORM:=LASTBUDWORM + 1;
    END;
  END;                                (* of budworm *)


PROCEDURE WEEVIL;
(*****************************************************************
 * -Called from SETPESTS                                         *
 * -Sets potential weevil levels for this year. Designed so that *
 *  they climb linearly for the first 10 years and hold steady   *
 *  afterwords. Weeviling is set to 0 for stands less than 3     *
 *  years old and for stands taller than 17 feet tall.           *
 *****************************************************************)
  VAR
    TEMPINT:INTEGER;
    TEMPREAL:REAL;

  BEGIN
    TEMPREAL:=1 - EXP(-0.02233*TREE_AGE);
    IF (TEMPREAL <= 0.0001) THEN TEMPREAL:=0.0001;
    HEIGHT:=1.633*SITE_INDEX*EXP(1.2419*LN(TEMPREAL));
    TEMPREAL:=1.0;
    IF (WEEVSPRAY) THEN BEGIN
      IF (RAN_OR_DET = 'Random') THEN TEMPINT:=RANDOM_INTEGER(20) + 65
      ELSE TEMPINT:=75;
      TEMPREAL:=TEMPINT/100;
      END;
    IF (COUNTYR>3) AND (HEIGHT<17) THEN
      IF (COUNTYR <= 10) THEN BEGIN
        WPW_WO_CONTROL:=FIRSTWEEV + (TREE_AGE*(LASTWEEV - FIRSTWEEV)/10.0);
        WPW_W_CONTROL:=TEMPREAL*WPW_WO_CONTROL;
        END
      ELSE BEGIN
        WPW_WO_CONTROL:=LAST_WEEV;
        WPW_W_CONTROL:=TEMPREAL*LAST_WEEV;
```

```
        END
      ELSE BEGIN
        WPW_WO_CONTROL:=0.0;
        WPW_W_CONTROL :=0.0;
        END;
      WEEVSPRAY:=FALSE;
      END;                          (* of weevil *)

BEGIN                               (* setpests *)
  WEEVIL;
  BUCWORM;
  END;                              (* of setpests *)
```

```
(* ********** MAIN.TEXT ********** *)


PROCEDURE GROWTHESTAND;
(*********************************************************************
 * -Called from main program BANKSIANA                               *
 * -Sets average height, basal area, and cords for the stand. Uses   *
 *  equations taken from Benzie J.W.,1977. Manager's handbook for    *
 *  jack pine in the North Central States. USDA For Serv Gen Tech    *
 *  Rep NC-32, 18 p.                                                 *
 *********************************************************************)
  VAR
    TEMPREAL:REAL;
    CH:CHAR;

  BEGIN
    TEMPREAL:=1 - EXP(-0.02233*TREE_AGE);
    IF (TEMPREAL <= 0.0001) THEN TEMPREAL:=0.0001;
    HEIGHT:=1.633*SITE_INDEX*EXP(1.2419*LN(TEMPREAL));

    TEMPREAL:=EXP(-4*EXP(-29.18/TREE_AGE));
    TEMPREAL:=TEMPREAL*0.276*EXP(0.63*LN(SITE_INDEX));
    TEMPREAL:=TEMPREAL*(1 + 0.056*BA_W_CONTROL -
         0.000358*BA_W_CONTROL*BA_W_CONTROL);
    IF (TEMPREAL<0) THEN TEMPREAL:=0;
    BA_W_CONTROL:=P_LIVE_W_CONTROL*(BA_W_CONTROL + P_GRO_W_CONTROL*TEMPREAL);
    CORDS_W_CONTROL:=0.003958*BA_W_CONTROL*HEIGHT;
    N_TRS_W_CONTROL:=N_TRS_W_CONTROL*P_LIVE_W_CONTROL;

    TEMPREAL:=EXP(-4*EXP(-29.18/TREE_AGE));
    TEMPREAL:=TEMPREAL*0.276*EXP(0.63*LN(SITE_INDEX));
    TEMPREAL:=TEMPREAL*(1 + 0.056*BA_WO_CONTROL -
         0.000358*BA_WO_CONTROL*BA_WO_CONTROL);
    IF (TEMPREAL<0) THEN TEMPREAL:=0;
    BA_WO_CONTROL:=P_LIVE_W_CONTROL*(BA_WO_CONTROL +
                                P_GRO_WO_CONTROL*TEMPREAL);
    CORDS_WO_CONTROL:=0.003958*BA_WO_CONTROL*HEIGHT;
    N_TRS_WO_CONTROL:=N_TRS_WO_CONTROL*P_LIVE_WO_CONTROL;
    END;                          (* of grow the stand *)

FUNCTION BUD_EFFECT(THIS_YR,LAST_YR,YR_BEFORE:REAL):REAL;
(*********************************************************************
 * -Called from EARLY_DAMAGE and LATE_DAMAGE                         *
 * -Looks up budworm damage on the table BUD_TABLE based on          *
 *  defoliation level and on the number of years since this          *
 *  defoliation occurred. Does this for last three year's defoliation *
 *  and multiplies results together to get total effect of budworm on *
 *  this year's tree growth.                                         *
 *********************************************************************)
  VAR
    ROW1,ROW2,ROW3:INTEGER;
    TEMP:REAL;
```

```
  BEGIN
    IF (THIS_YR <= 3) THEN ROW1:=1
    ELSE IF (THIS_YR <= 5) THEN ROW1:=2
    ELSE IF (THIS_YR <= 7) THEN ROW1:=3
    ELSE IF (THIS_YR = 8) THEN ROW1:=4
    ELSE ROW1:=5;
    IF (LAST_YR <= 3) THEN ROW2:=1
    ELSE IF (LAST_YR <= 5) THEN ROW2:=2
    ELSE IF (LAST_YR <= 7) THEN ROW2:=3
    ELSE IF (LAST_YR = 8) THEN ROW2:=4
    ELSE ROW1:=5;
    IF (YR_BEFORE <= 3) THEN ROW3:=1
    ELSE IF (YR_BEFORE <= 5) THEN ROW3:=2
    ELSE IF (YR_BEFORE <= 7) THEN ROW3:=3
    ELSE IF (YR_BEFORE = 8) THEN ROW3:=4
    ELSE ROW1:=5;
    BUD_EFFECT:=BUD_TABLE[ROW1,1]*BUD_TABLE[ROW2,2]*BUD_TABLE[ROW3,3];
    END;                          (* of bud effect *)

PROCEDURE SETDAMAGE;
(*************************************************************************
 * -Called from main program BANKSIANA                                   *
 * -Based on insect attack levels, sets amount of mortality, amount      *
 *  tree growth, and proportion of trees rendered unmarketable.          *
 *  For stands younger than 20 years, damage is accumulated for          *
 *  eventual set up of an initial stand at year 20. Damage for older     *
 *  stands are applied to the stand the year that they occur.            *
 *************************************************************************)
  VAR
    JPBWMORTALITY:REAL;
    TEMPREAL:REAL;

  PROCEDURE EARLY_DAMAGE;
  (*********************************************************************
   * -Called from SET_DAMAGE                                           *
   * -Sets growth loss based on jack pine populations and BUD_TABLE    *
   *  Sets mortality based on jack pine populations over the last      *
   *  years. Mortality function is designed so that mortality is at    *
   *  one percent except in extreme population levels where it can     *
   *  rise as high as 90 percent. Proportion of trees hit once, twice  *
   *  etc. by the weevil are determined. Trees hit more than four      *
   *  times are considered unmarketable.                               *
   *********************************************************************)
    BEGIN
                                                  (*  growth loss *)
      TEMPREAL:=BUD_EFFECT(BW1_W_CONTROL,BW2_W_CONTROL,BW3_W_CONTROL);
      P_GRO_W_CONTRO:=P_GRO_W_CONTROL + TEMPREAL;
      TEMPREAL:L=BUD_EFFECT(BW1_WO_CONTROL,BW2_WO_CONTROL,BW3_WO_CONTROL);
      P_GRO_WO_CONTROL:=P_GRO_WO_CONTROL + TEMPREAL;
                                                   (* mortality *)
     TEMPREAL:=BW1_W_CONTROL  + BW2_W_CONTROL + BW3_W_CONTROL;
      IF (TEMPREAL < 20) THEN TEMPREAL:=20.0;
      TEMPREAL:=-1.563 + 0.0940*TEMPREAL;
```

```
    JPBWMORTALITY:=TEMPREAL*TEMPREAL*TEMPREAL*TEMPREAL;
    IF (JPBWMORTALITY < 0) THEN JPBWMORTALITY:=0.0;
    TEMPREAL:=1 - JPBWMORTALITY;
    P_LIVE_W_CONTROL:=P_LIVE_W_CONTROL*TEMPREAL;
    TEMPREAL:=BW1_WO_CONTROL + BW2_WO_CONTROL + BW3_WO_CONTROL;
    IF (TEMPREAL <= 20) THEN TEMPREAL:=20;
    TEMPREAL:=-1.563 + 0.0940*TEMPREAL;
    JPBWMORTALITY:=TEMPREAL*TEMPREAL*TEMPREAL*TEMPREAL;
    IF (JPBWMORTALITY < 0) THEN JPBWMORTALITY:=0.0;
    TEMPREAL:=1 - JPBWMORTALITY;
    P_LIVE_WO_CONTROL:=P_LIVE_WO_CONTROL*TEMPREAL;
                                        (* weevil damage *)
    TEMPREAL:=WPW_W_CONTROL/100;
    HIT_4X_W_CONTROL:=HIT_4X_W_CONTROL + TEMPREAL*HIT_3X_W_CONTROL;
    HIT_3X_W_CONTROL:=HIT_3X_W_CONTROL +
         TEMPREAL*(HIT_2X_W_CONTROL - HIT_3X_W_CONTROL);
    HIT_2X_W_CONTROL:=HIT_2X_W_CONTROL +
         TEMPREAL*(HIT_1X_W_CONTROL - HIT_2X_W_CONTROL);
    HIT_1X_W_CONTROL:=HIT_1X_W_CONTROL +
         TEMPREAL*(HIT_0X_W_CONTROL - HIT_1X_W_CONTROL);
    HIT_0X_W_CONTROL:=HIT_0X_W_CONTROL -
         TEMPREAL*HIT_0X_W_CONTROL;
    TEMPREAL:=WPW_WO_CONTROL/100;
    HIT_4X_WO_CONTROL:=HIT_4X_WO_CONTROL + TEMPREAL*HIT_3X_WO_CONTROL;
    HIT_3X_WO_CONTROL:=HIT_3X_WO_CONTROL +
         TEMPREAL*(HIT_2X_WO_CONTROL - HIT_3X_WO_CONTROL);
    HIT_2X_WO_CONTROL:=HIT_2X_WO_CONTROL +
         TEMPREAL*(HIT_1X_WO_CONTROL - HIT_2X_WO_CONTROL);
    HIT_1X_WO_CONTROL:=HIT_1X_WO_CONTROL +
         TEMPREAL*(HIT_0X_WO_CONTROL - HIT_1X_WO_CONTROL);
    HIT_0X_WO_CONTROL:=HIT_0X_WO_CONTROL -
         TEMPREAL*HIT_0X_WO_CONTROL;
    END;                                (* of early damage *)

PROCEDURE LATE_DAMAGE;
(*********************************************************************
 * -Called from SET_DAMAGE                                           *
 * -Sets growth loss based on jack pine populations and BUD_TABLE    *
 *  Sets mortality based on jack pine populations over the last      *
 *  years. Mortality function is designed so that mortality is at    *
 *  one percent except in extreme population levels where it can     *
 *  rise as high as 33 percent. Weevil no longer is considered.      *
 *********************************************************************)
  BEGIN
    P_GRO_W_CONTROL:=BUD_EFFECT(BW1_W_CONTROL,BW2_W_CONTROL,BW3_W_CONTROL);
    P_GRO_WO_CONTROL:=BUD_EFFECT(BW1_WO_CONTROL,BW2_WO_CONTROL,
         BW3_WO_CONTROL);
    TEMPREAL:=BW1_W_CONTROL + BW2_W_CONTROL + BW3_W_CONTROL;
    IF (TEMPREAL < 20) THEN TEMPREAL:=20;
    TEMPREAL:=-0.946 + 0.0631*TEMPREAL;
    JPBWMORTALITY:=TEMPREAL*TEMPREAL*TEMPREAL*TEMPREAL;
    IF (JPBWMORTALITY < 0) THEN JPBWMORTALITY:=0.0;
    P_LIVE_W_CONTROL:=1 - JPBWMORTALITY;
    TEMPREAL:=BW1_WO_CONTROL + BW2_WO_CONTROL + BW3_WO_CONTROL;
```

```
      IF (TEMPREAL < 20) THEN TEMPREAL:=20.0;
      TEMPREAL:=-0.946 + 0.0631*TEMPREAL;
      JPBWMORTALITY:=TEMPREAL*TEMPREAL*TEMPREAL*TEMPREAL;
      IF (JPBWMORTALITY < 0) THEN JPBWMORTALITY:=0.0;
      P_LIVE_WO_CONTROL:=1 - JPBWMORTALITY;
      END;                              (* of late damage *)


  BEGIN
    IF (TREE_AGE<20) THEN EARLY_DAMAGE
    ELSE LATE_DAMAGE;
    END;                              (* of set damage *)


FUNCTION BASAL_AREA (VAR NUM_TREES,PROP_LIV,PROP_GRO:REAL;
                                        SITE_INDEX:INTEGER):REAL;
(*********************************************************************
 * -Called from main program BANKSIANA                               *
 * -Sets up initial basal area of the stand based on table 2 of the  *
 *  jack pine manager's handbook (cited in GROW_THE_STAND). The      *
 *  of trees is cross referenced with the site index to give initial *
 *  basal area. If number of trees is less than what is on the table *
 *  then a Lagrange Interpolation is used to pick some value between *
 *  the lowest tree number values listed on the table and the case   *
 *  when no trees are growing.                                       *
 *********************************************************************)
  VAR
    BACHART:ARRAY[1..3,1..4] OF REAL;
    K,L:INTEGER;
    EST1:REAL;
    EST2:REAL;
    EST3:REAL;
    EST4:REAL;
    LOWER_SI:REAL;
    LOWER_NUM_TREES:REAL;
    TEMPREAL:REAL;


  BEGIN
                      (* Table 2, Jack pine manager's handbook *)
    BA_CHART[3,1]:=39.0;
    BA_CHART[3,2]:=63.0;
    BA_CHART[3,3]:=79.0;
    BA_CHART[3,4]:=92.0;
    BA_CHART[2,1]:=28.0;
    BA_CHART[2,2]:=46.0;
    BA_CHART[2,3]:=55.0;
    BA_CHART[2,4]:=63.0;
    BA_CHART[1,1]:=17.0;
    BA_CHART[1,2]:=29.0;
    BA_CHART[1,3]:=36.0;
    BA_CHART[1,4]:=39.0;
    NUM_TREES:=(8/9)*PROP_LIV*NUM_TREES;
    K:=TRUNC((NUM_TREES + 1)/500);
```

```
      IF (K<1) THEN K:=1;
      L:=TRUNC((SITE_INDEX + 1)/10 - 3);
      IF (L<1) THEN L:=1;
      LOWER_NUM_TREES:=TRUNC((NUM_TREES + 1)/500)*500;
      IF (LOWER_NUM_TREES<500) THEN LOWER_NUM_TREES:=500;
      LOWER_SI:=TRUNC((SITE_INDEX + 1)/10)*10;
      IF (LOWER_SI < 40) THEN LOWER_SI:=40;
      IF (NUM_TREES<500) THEN
                                    (* Lagrange interpolation *)
        TEMPREAL:=(0.0022*NUM_TREES*SITE_INDEX - 0.054*NUM_TREES)*PROP_GRO
      ELSE IF (NUM_TREES - LOWER_NUM_TREES <0.001) THEN
        IF (SITE_INDEX>59) THEN TEMPREAL:=(BA_CHART[3,K])*PROP_GRO
        ELSE TEMPREAL:=PROP_GRO*(BA_CHART[L,K] +
          ((SITE_INDEX - LOWER_SI)/10)*(BA_CHART[L+1,K] - BA_CHART[L,K]))
      ELSE IF (SITE_INDEX - LOWER_SI < 0.001) THEN
        IF (NUM_TREES>1999) THEN TEMPREAL:=BA_CHART[K,4]*PROP_GRO
        ELSE TEMPREAL:=PROP_GRO*(BA_CHART[L,K] +
          ((NUM_TREES - LOWER_NUM_TREES)/500)*(BA_CHART[L,K+1] - BA_CHART[L,K]))
      ELSE BEGIN
        EST1:=BA_CHART[L,K] +
              ((SITE_INDEX - LOWER_SI)/10)*(BA_CHART[L+1,K]- BA_CHART[L,K]);
        EST2:=BA_CHART[L+1,K] + ((NUM_TREES - LOWER_NUM_TREES)/500)
                              *(BA_CHART[L+1,K+1] - BA_CHART[L+1,K]);
        EST3:=BA_CHART[L,K+1] + ((SITE_INDEX - LOWER_SI)/10)
                              *(BA_CHART[L+1,K+1] - BA_CHART[L,K+1]);
        EST4:=BA_CHART[L,K] + ((NUM_TREES - LOWER_NUM_TREES)/500)
                              *(BA_CHART[L,K+1] - BA_CHART[L,K]);
        TEMPREAL:=((EST1 + EST2 + EST3 + EST4)/4)*PROP_GRO;
        END;
      BASAL_AREA:=TEMPREAL;
      END;                    (* of init basal area *)


PROCEDURE DECIDE_CHECK;
(*********************************************************************
 * -Called from main program BANKSIANA                               *
 * -If the program is running automatically, this procedure          *
 *  determines whether or not to monitor or manage this year based on *
 *  what options within automatic run were chosen and whether or not  *
 *  those options call for monitoring/managing.                      *
 *********************************************************************)
  VAR
    A,B,C,D,E:BOOLEAN;
    OK_DO_BW,OK_DO_WV:BOOLEAN;

  BEGIN
    A:=FALSE;
    B:=FALSE;
    C:=FALSE;
    D:=FALSE;
    E:=FALSE;
    IF (COUNTYR>=START_BW_PLAN) AND (COUNTYR<=END_BW_PLAN) THEN
         OK_DO_BW:=TRUE
    ELSE OK_DO_BW:=FALSE;
```

```
IF (COUNTYR>=START_WV_PLAN) AND (COUNTYR<=END_WV_PLAN) THEN
        OK_DO_WV:=TRUE
    ELSE OK_DO_WV:=FALSE;
    IF (BW_PLAN = 'Spray') AND (NEXT_BUDWORM = 0) AND OK_DO_BW
        THEN A:=TRUE;
    IF (BW_PLAN = 'MonSpray') AND (BW_MON_PLAN = 'Every ?? yrs') AND
        (BW_CHECK = BUD_COUNT) AND OK_DO_BW THEN B:=TRUE;
    IF (BW_PLAN = 'MonSpray') AND (BW_MON_PLAN = 'During outbrks') AND
        (NEXT_BUDWORM = 0) AND OK_DO_BW THEN C:=TRUE;
    IF (WEEV_PLAN = 'Spray') AND (WEEV_CHECK = WEEV_COUNT) AND OK_DO_WV
        THEN D:=TRUE;
    IF (WEEV_PLAN = 'MonSpray') AND (WEEV_CHECK = WEEV_COUNT)
        AND OK_DO_WV THEN E:=TRUE;
    IF A OR B OR C OR D OR E OR CUT THEN CHECK_STAND;
    END;                                (* of decide check *)

BEGIN                       (* Main Program! *)
   INITIALIZE;
   INIT_BIOL;
   REPEAT
     SETPESTS;
     IF (AUTO) THEN DECIDE_CHECK
     ELSE IF (CHECK<=0) THEN CHECKSTAND;
     SETDAMAGE;
     CHECK:=CHECK - 1;
     IF (CHECK<0) THEN CHECK:=0;
     TREE_AGE:=TREE_AGE + 1;
     IF AUTO THEN
       IF (TREE_AGE >= START_WV_PLAN) AND (TREE_AGE <= END_WV_PLAN) THEN
           WEEV_COUNT:=WEEV_COUNT + 1;
     COUNTYR:=COUNTYR + 1;
     PAGE(OUTPUT); GOTOXY(0,8);
     WRITELN('                              Year ',COUNTYR:2);
  UNTIL(TREE_AGE>=20);
  WEEV_COUNT:=0;
  IF (CUT) THEN WRITELN
  ELSE BEGIN
    P_GRO_W_CONTROL:=P_GRO_W_CONTROL/18;
    P_GRO_WO_CONTROL:=P_GRO_WO_CONTROL/18;
    BA_W_CONTROL:=BASAL_AREA(N_TRS_W_CONTROL,P_LIVE_W_CONTROL,
                    P_GRO_W_CONTROL,SITE_INDEX);
    BA_WO_CONTROL:=BASAL_AREA(N_TRS_WO_CONTROL,P_LIVE_WO_CONTROL,
                    P_GRO_WO_CONTROL,SITE_INDEX);
    REPEAT
      SETPESTS;
      IF AUTO THEN BEGIN
        IF (TREE_AGE = ROTATION) THEN CUT:=TRUE;
        DECIDE_CHECK;
        END
      ELSE IF (CHECK<=0) THEN CHECKSTAND;
      SETDAMAGE;
      GROWTHESTAND;
      CHECK:=CHECK - 1;
      IF (CHECK<0) THEN CHECK:=0;
```

```
  TREE_AGE:=TREE_AGE + 1;
  IF AUTO THEN BUD_COUNT:=BUD_COUNT + 1;
  COUNTYR:=COUNTYR +1;
  PAGE(OUTPUT); GOTOXY(0,8);
  WRITELN('                                Year ',COUNTYR:2);
  UNTIL (COUNTYR>=70) OR (FINISHED);
IF (TREE_AGE>=70) THEN BEGIN
    WRITE('Your stand is at least 70 years old.');
    WRITELN('..... TIME TO CUT!!!!!!!!');
    GOTOXY(0,20);WRITELN('              RETURN');
    READLN(CH);
    MANAGE;
    END;
```

APPENDIX B

Text Files Associated with the Program

Figure B1. The text from file 'HELLO.TEXT.'

```
   JACK PINE MANAGEMENT MODEL -- INITIALIZATION PHASE

 1) Type of user: ----------------

 2) Type of run: ----------

 3) Number of acres (10 - 999): ---

 4) Site index (45 - 60): --

 5) Discount rate % (0.0 - 15.0): --.--

 6) Interactive vs. Automatic run.   (Default = interactive)

 7) Initial weevil density: --------

 8) Initialize management costs

 9) Select site preparation and planting

10) Run model

11) Exit program
```

Figure B2. The text from file 'MASK1.TEXT'

```
        PACK PINE MANAGEMENT MODEL -- MANAGEMENT COSTS
-------------------------------------------------------------------------
Initialize wages of personnel.
       1) Hourly wage of pest scout:  --.--

       2) Hourly wage of forest technician:  --.--

Initialize management costs for pesticide application.
   Applications targeted against Jack Pine Budworm
       3) Aerial, chemical spray:  --.-- +  --.-- per acre =  ----.--

       4) Ground, chemical spray:  --.-- +  --.-- per acre =  ----.--

       5) Aerial, biological spray:--.-- +  --.-- per acre =  ----.--

       6) Ground, biological spray:--.-- +  --.-- per acre =  ----.--

   Applications targeted against White Pine Weevil
       7) Aerial, chemical spray:  --.-- +  --.-- per acre =  ----.--

       8) Ground, chemical spray:  --.-- +  --.-- per acre =  ----.--


     9) Go back to the initialization menu
    10) Exit program
```

Figure B3. The text from file 'MASK2.TEXT'

JACK PINE MANAGEMENT MODEL -- SITE PREPARATION AND REGENERATION

1) Burn: ---

2) Herbicide: ---

3) Lop and scatter: ---

4) Scarify: ---

5) Rollerchop: ---

6) Method of regeneration: ----------

7) Go back to the initialization menu

Figure B4. The text from file 'MASK3.TEXT'

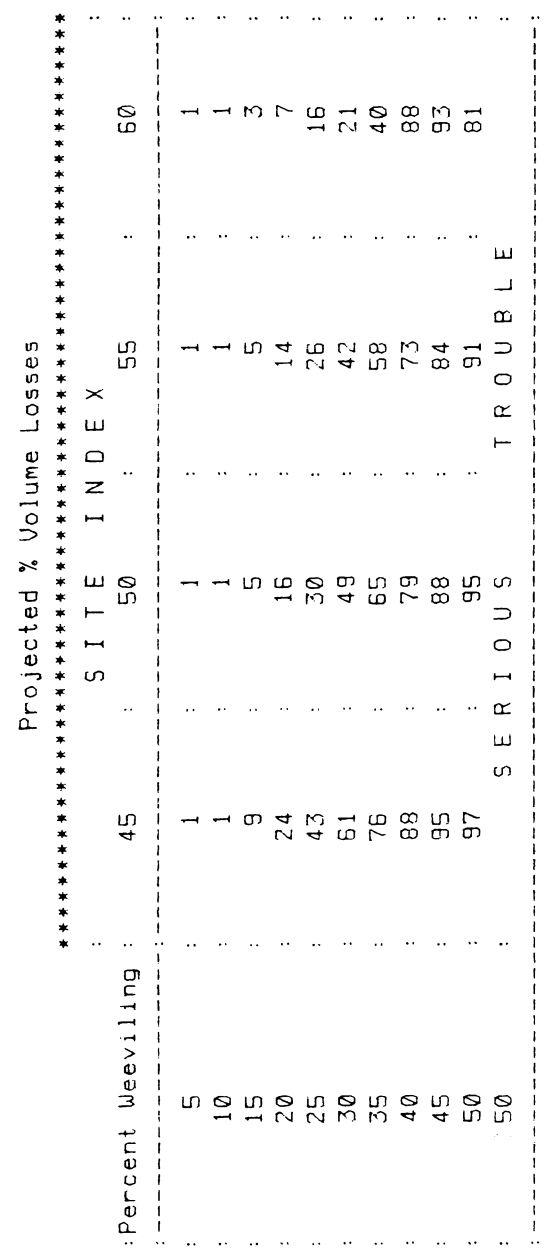


Figure B5. The text from file 'WEEVCHART.TEXT'

APPENDIX C

Library Procedures used in the Program

```
PROCEDURES BORROWED FROM UNIT "UTILITIES."*


PROCEDURE PAUSE(* SECONDS:INTEGER *);

CONST ONE_SECOND = 3600;

VAR I,J: INTEGER;

BEGIN
  IF SECONDS > 0 THEN
    FOR I := 1 TO SECONDS DO
      FOR J:= 1 TO ONE_SECOND DO
END;


PROCEDURE BEEP;

BEGIN
  WRITE(CHR(BELL));
END;



PROCEDURE CAP(* VAR CH:CHAR *)

BEGIN
  IF CH IN ['a'..'z'] THEN
    CH:=CHR(ORD(CH) - ORD('a') + ORD('A');
END;


PROCEDURE  FILL_FIELD(* X,Y FIELD_LENGTH:INTEGER; CH:CHAR *)

VAR I:INTEGER;

BEGIN
  GOTOXY(X,Y);
  IF FIELD_LENGTH > 0 THEN
    FOR I:=1 TO FIELD_LENGTH DO
      WRITE(CH);
  GOTOXY(X,Y);
END;
```

*Version 3.0. Released August 27, 1984. Programmers: Edward Dawson, Stuart Gage, Robert Kriegel, and Aubrey Moore.

```
PROCEDURE STR_TO_INT(* VAR NUMBER:INTEGER; STR:STRING;
                       VAR OK:BOOLEAN*)
VAR I:INTEGER;
    NEGATIVE:BOOLEAN;

BEGIN
  NUMBER:=0;
  OK:=TRUE; NEGATIVE:=FALSE;
  IF LENGTH(STR) = 0 THEN
    OK:=FALSE;
  ELSE
    BEGIN
      FOR I:= 1 TO LENGTH(STR) DO
        BEGIN
          IF STR[I] IN ['0'..'9'] THEN
            BEGIN
              NUMBER:= NUMBER * 10;
              NUMBER:= NUMBER + ORD(STR[I]) - ORD('0');
            END
          ELSE IF (STR[I] = '.') AND (I = LENGTH(STR)) THEN
          ELSE IF (I=1) AND (STR[I] IN ['+','-'] THEN
            BEGIN
              IF STR[1] = '-' THEN
                NEGATIVE:=TRUE;
            END
          ELSE
            OK:= FALSE;
        END;
      IF NEGATIVE THEN
        NUMBER:=-1 * NUMBER;
    END;
END;


PROCEDURE STR_TO_REAL(* VAR NUMBER:REAL; STR:STRING; VAR OK:BOOLEAN *);

VAR ANY_DECIMALS, NEGATIVE, ANY_EXPONANT, OK_TEMP:BOOLEAN;
  I, COUNTER, TEMP, EXPONANT,:INTEGER;
  WHOLE, FRACTION, POWER, TEMP_STR:STRING;
  INT, DECIMAL:REAL;

BEGIN
  OK_TEMP:=TRUE; DECIMAL:=0.0; TEMP:=0;
  IF LENGTH(FRACTION) > 0 THEN
    FOR I:= LENGTH(FRACTION) DOWNTO 1 DO
      BEGIN
        TEMP_STR[1] := FRACTION[I];
        STR_TO_INT(TEMP,TEMP_STR,OK_TEMP);
        DECIMAL:=(DECIMAL + TEMP) / 10.0;
        IF NOT OK_TEMP THEN
          OK:=FALSE;
```

```
      END;
    NUMBER:=INT + DECIMAL;
    IF NEGATIVE THEN
      NUMBER:=-1.0*NUMBER;
  OK_TEMP:=TRUE; EXPONENT:=0; TEMP:=0;
  IF ANY_EXPONENT THEN
    BEGIN
      STR_TO_INT(EXPONENT,POWER,OK_TEMP);
      IF NOT OK_TEMP THEN
        OK:= FALSE;
      IF EXPONENT:=0 THEN
        NUMBER:=1.0
      ELSE
        NUMBER:= NUMBER*EXP(EXPONENT*LN(10.0));
    END;
  END;
END;


PROCEDURE WRITE_INT(* X,Y,NUMBER, TOTAL_FIELD:INTEGER *);

BEGIN
  GOTOXY(X,Y);
  WRITE(NUMBER:TOTAL_FIELD);
END;


PROCEDURE INT_REPLACE(* VAR NUMBER:INTEGER; X,Y, TOTAL_FIELD:
                        INTEGER *);
VAR STR:STRING;
    OLD_NUMBER:INTEGER;
    DONE, OK:BOOLEAN;

BEGIN
  WRITE_INT(X,Y,NUMBER,TOTAL_FIELD);
  OLD_NUMBER:=NUMBER;
  DONE:=FALSE;
  REPEAT
    GOTOXY(X,Y);
    READLN(STR);
    FILL_FIELD(X,Y,TOTAL_FIELD,' ');
    IF LENGTH(STR) = 0 THEN
      BEGIN
        NUMBER:=OLD_NUMBER;
        DONE:=TRUE;
      END
    ELSE IF LENGTH(STR) IN [1..TOTAL_FIELD] THEN
      BEGIN
        STR_TO_INT(NUMBER,STR,OK);
        IF OK THEN
```

```
          DONE:=TRUE;
      END;
    IF NOT DONE THEN
      BEGIN
        WRITE('?',' ':(TOTAL_FIELD - 1));
        BEEP;
        PAUSE(1);
      END;
  UNTIL DONE;
  WRITE_INT(X,Y,NUMBER,TOTAL_FIELD);
  WRITELN;
END;



PROCEDURE REAL_REPLACE(*VAR NUMBER:REAL; X,Y, TOTAL_FIELD,
                  DECIMAL_FIELD: INTEGER *);
VAR STR:STRING;
    OLD_NUMBER:REAL;
    DONE, OK: BOOLEAN;

BEGIN
  WRITE_REAL(X,Y,NUMBER,TOTAL_FIELD,DECIMAL_FIELD);
  OLD_NUMBER:=NUMBER;
  DONE:= FALSE;
  REPEAT
    GOTOXY(X,Y);
    READLN(STR);
    FILL_FIELD(X,Y,TOTAL_FIELD,' ');
    IF LENGTH(STR) = 0 THEN
      BEGIN
        NUMBER:= OLD_NUMBER;
        DONE:=TRUE;
      END
    ELSE IF LENGTH(STR) IN [1..TOTAL_FIELD] THEN
      BEGIN
        STR_TO_REAL(NUMBER,STR,OK);
        IF OK THEN
          DONE:= TRUE;
      END
    IF NOT DONE THEN
      BEGIN
        WRITE('?',' ':(TOTAL_FIELD - 1));
        BEEP;
        PAUSE(1);
      END;
  UNTIL DONE;
  WRITE_REAL(X,Y,NUMBER,TOTAL_FIELD,DECIMAL_FIELD);
  WRITELN;
END;
```

```
PROCEDURE WRITE_STRING(* X,Y:INTEGER; STR:STRING; TOTAL_FIELD:
                                              INTEGER *);
BEGIN
  GOTOXY(X,Y):
    WRITE(STR:TOTAL_FIELD);
END;



PROCEDURE WRITE_LONG_STRING(* X,Y:INTEGER; LONG:SC_LONG_STRING;
                            BREAK_CHAR:CHAR *);
VAR I:INTEGER;

BEGIN
  GOTOXY(X,Y);
  IF LENGTH(LONG) > 0 THEN
    FOR I:=1 TO LENGTH(LONG) DO
      IF LONG[I] = BREAK_CHAR THEN
        BEGIN
          Y:=Y + 1;
          GOTOXY(X,Y);
        END
      ELSE
        WRITE(LONG[I]);
END;



FUNCTION CHAR_REPLACE(* X,Y:INTEGER; RETURN_ON_MATCH:SC_CHSET)
                                    :CHAR *);
VAR DUMMY_CH:CHAR;

BEGIN
  GOTOXY(X,Y);
  SC_GETC_CH(DUMMY_CH, RETURN_ON_MATCH);
  WRITE(DUMMY_CH);
  CHAR_REPLACE:=DUMMY_CH;
END;



PROCEDURE CH_PROMPT(* X,Y:INTEGER; PROMPT:SC_LONG_STRING;
           BREAK_CHAR:CHAR; NEW_X, NEW_Y:INTEGER;
           RETURN_ON_MATCH:SC_CHSET):CHAR *);
BEGIN
  SC_ERASE_TO_EOL(X,Y);
  WRITE_LONG_STRING(X,Y,PROMPT,BREAK_CHAR);
  CH_PROMPT := CHAR_REPLACE(NEW_X, NEW_Y, RETURN_ON_MATCH);
END;
```

```
PROCEDURE INT_PROMPT(* X,Y:INTEGER; PROMPT:SC_LONG_STRING;
          BREAK_CHAR:CHAR; NEW_X, NEW_Y:INTEGER;
          VAR NUMBER:INTEGER; TOTAL_FIELD:INTEGER *);
BEGIN
  SC_ERASE_TO_EOL(X,Y);
  WRITE_LONG_STRING(X,Y,PROMPT,BREAK_CHAR);
  INT_REPLACE(NUMBER,NEW_X,NEW_Y,TOTAL_FIELD);
END:



PROCEDURE REAL_PROMPT(* X,Y:INTEGER; PROMPT:SC_LONG_STRING;
          BREAK_CHAR:CHAR; NEW_X, NEW_Y:INTEGER;
          VAR NUMBER:REAL; TOTAL_FIELD, DECIMAL_FIELD:INTEGER *);
BEGIN
  SC_ERASE_TO_EOL(X,Y);
  WRITE_LONG_STRING(X,Y,PROMPT,BREAK_CHAR);
  REAL_REPLACE( NUMBER, NEW_X, NEW_Y, TOTAL_FIELD, DECIMAL_FIELD);
END;



PROCEDURE SCREED_MASK(* FILE_NAME:STRING;VAR OK:BOOLEAN *);

VAR A_FILE:TEXT;
    IORES:INTEGER;
    ONE_LINE:STRING;
    CONSOLE:INTERACTIVE;

BEGIN
  SC_CLR_SCREEN;
  GOTOXY(0,0);
(*$I-*)
  REWRITE(CONSOLE,'CONSOLE:');
  RESET(A_FILE,FILE_NAME);
  IORES:=IORESULT;
  IF IORES = 0 THEN
    BEGIN
      OK:=TRUE;
      WHILE NOT EOF(A_FILE) DO
        BEGIN
          READLN(A_FILE,ONE_LINE);
          WRITELN(CONSOLE,ONE_LINE);
        END;
    END
  ELSE BEGIN
    OK:=FALSE;
    WRITELN('Error in writing screen mask: ', FILE_NAME);
  END:
  CLOSE(A_FILE);
  CLOSE(CONSOLE,LOCK);
  (*$I=*)
END;
```