Sebastian Hall
CSC 5

Hammurabi: Table Of Contents

Introduction

The game Hammurabi, alternatively known as The Sumer Game, originated in 1968 as a text based computer game. Hammurabi focuses on the aspect of proper resource management, or conversely dealing with the consequences when there is a lack of management, and dealing with unexpected events provided by the game environment. The game is a single player game and requires strategy to end the game with higher standings than when the game began.

Game Rules

1.) The game lasts 10 years, with a year being one turn.

2.) Each year, enter how many bushels of grain to allocate to buying (or selling) acres of land, feeding your population, and planting crops for the next year.

3.) Each person needs 20 bushels of grain each year to live and can till at most 10 acres of land.

4.) Each acre of land requires one bushel of grain to plant seeds.

5.) The price of each acre of land fluctuates from 17 bushels per acre to 26 bushels.

6.) If the conditions in your country ever become bad enough, the people will overthrow you and you won't finish your 10-year term.

7.) If you make it to the 11th year, your rule will be evaluated and you'll be ranked against great figures in history.

Game Objectives

The objective of the game is to rule your land as the Babylonian king Hammurabi. Leave your land in a better condition than when you started at the end of the 10-year term the game lasts. Feed all of your people and attempt to purchase excess land to increase your holdings under your rule. Plant all your crops when to increase your yield to have the most favorable conditions and beware of the rats.

Pseudocode

```
Opening comments
System Libraries
User Libraries
Global constants
Function prototypes
Begin Main Function
    Set Random Seed
    Define Constants and Variables
    Do the following
        Call Display title function
            Define variables and file objects
            Open title file
            Read title to file
            Close file
            Open input file
            Read input file as title
            Close title file
            End line
            End Display title function
        Call Display Rules Function
            Define Variables
            Output options to see rules or not
            If chose to see rules display rules and press enter to play,
            End line and end see rules function
        Declare and initialize all the variables
        Call function to display the year’s data
            Output the yearly data
            End display data
        For all the years the game is played do the following
            Output question one
            Input answer
            Validate answer
            Update and display decision making info
            Output next question
            Read in answer
            Validate answer
            Fill array ttlFed sub year minus 2
            Update and output decision making info
            Call first loss function
                Declare variable to hold starved people
                If the food amount fed is not enough
                    Calculate how many people died
                Else starved people equals zero
                Return starved people
            Fill array deadYr sub 0 sub year minus 2 with year minus 1
            Fill array deadYr sub 1 sub year minus 2 starved people
            Output question three
            Read in and validate answer
            Update variable values for the next iteration
            If more than 45% died end the game and move to switch menu
            Call land price function
                Set random seed
```

```
                    Get random land price
                    Return land price
                    End land price function
          Call new population function
                    Set random seed
                    Get random new population
                    Return population
                    End new population function
          Call crops per acre function
                    Set random seed
                    Get random new crops
                    Return random crop value
                    End function
          Call second loss function
                    Set random seed
                    Get possibility of getting rats
                    If possibility is two then rats take up to 50% of the food
                    Else no food eaten by rats
                    Return eaten
                    End rat food function
                    Process more data for the next round
                    Fill rats vector sub year minus 2 with ratFood
                    Declare plague
                    If plague equals preset value
                              Take plague deaths out of population
                              Output message about plague deaths
          Call function to display the year's data
                    Output the yearly data
                    End display data
          For Loop ends or reiterates
If you finish with no land you lose
Else if you finish without killing everybody then you survived and output victory
message
          Output final statistics
```

```
            Ternary operator for deciding if stats are good enough to be compared
            with great leaders or not
    Call stats function
        Declare variables to hold totals and highest values
        For loop from i=0 to i<2
            For loop from j=0 to j<10
                Output the two dimensional array for how many died each year
            If I is 0 then output the literal "dead: "
        Define variables for sorting
        Do
            Set swap variable to false
            For loop to sort the deadYr array to find the most killed in one year
        While swap is true
        For loop to calculate total of people fed
        For loop to calculate the total grains grown
        For loop to calculate how much was eaten by rats
        Output the highest killed, total grains eaten, and total eaten by rats
        End stats function
    Asks user if they want to play again or quit
    Input answer
    If answer invalid
        Say answer is invalid and prompt another input
            If answer is invalid again
                Specify the answer type asked for
    Switch base on answer received
        Case 1 display replay the game. Break away from switch
        Case 2 display game ending and end line
  End do. While answer for play again does not indicate quit
End main
```

Program

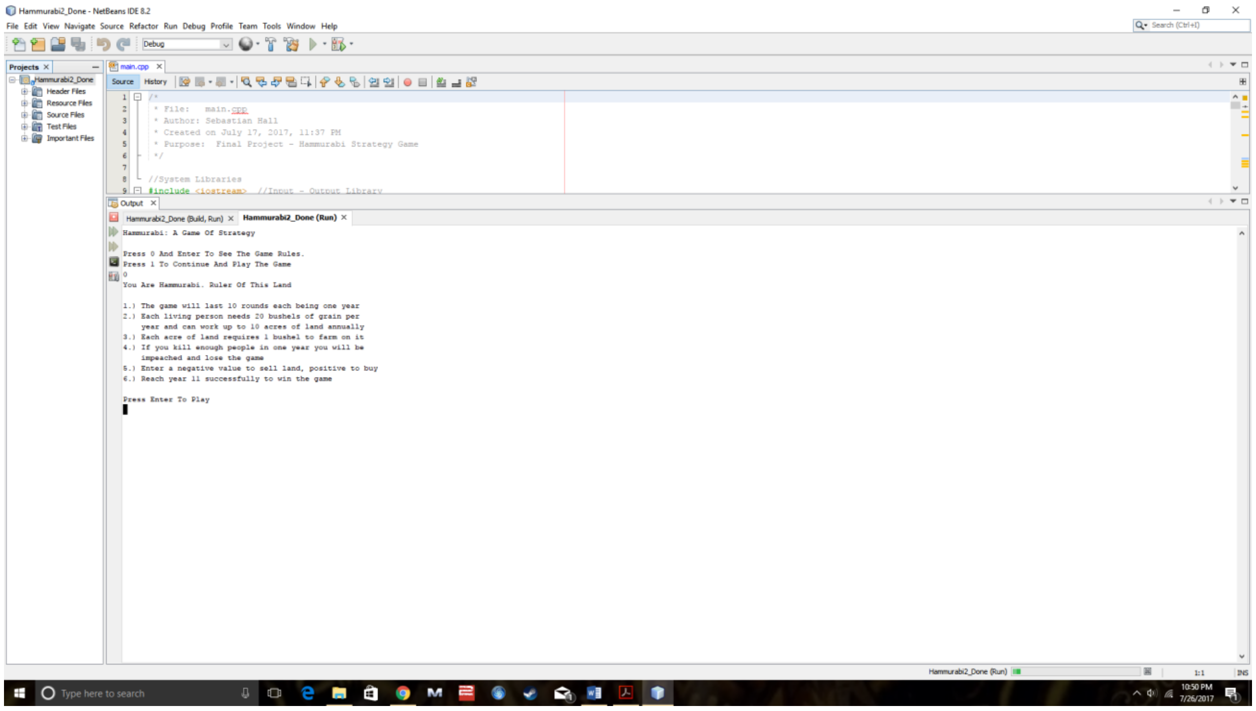
```
        /*
 * File:   main.cpp
 * Author: Sebastian Hall
 * Created on July 17, 2017, 11:37 PM
 * Purpose:  Final Project - Hammurabi Strategy Game
 */

//System Libraries
#include <iostream>  //Input - Output Library
#include <ctime>     //For Time Function
#include <cstdlib>   //For Rand and Srand
#include <fstream>   //For File Input/Output
#include <iomanip>   //Formatting
#include <cmath>     //For the math functions
#include <vector>    //For vector requirement
using namespace std; //Name-space under which system libraries exist

//User Libraries

//Global Constants
short const ENDYR=11;//The year constant for arrays

//Function Prototypes
void gtTitle();//Output title using file input/output
void seeRule();//Letting the player see the rules of the game
void dspYear(int &,float ,int ,float &,int &,int &,int ,int ,int);//Display
                                    //the status for the current year
short priceL();     //New price each year
short neoPop();     //The new population each year
short cropRnd();//Random crop growth each year
short loss(int);//Bushels lost by rats     //Functions Overloading
int loss(float ,int);//People lost by starvation
void stats(int [][ENDYR],int [],int [],vector<int>);//To output stats

//Execution begins here
int main() {
    //Setting random seed
    srand(static_cast<unsigned int>(time(0)));

    //Define menu choice variable
    short plyAgn;//Play Again?
```

```
//Do while to loop for replays
do{

//Display The Title
gtTitle();
cout<<endl;

//Optional Rules To See
seeRule();

//Declare and initialize variables
static int year=1;//The first year
float newPpl=0;//New people to be determined by random
int strvd=0;//The number of people you failed to feed and killed
float pop=100;//The city population. Starts at 100
int acres=1000;//City starts with 1000 acres
int totBush=2800;//Total bushels starting at 2800
int crops=0;//Amount harvested each year
int ratFood=0;//The amount the rats happened to eat that year
int lndPrc=rand()%10+17;//The current going rate for land in bushels range [17,26]
short sellBuy=0;//The number of acres one wishes to sell or buy
short acrsWrk=0;//The amount of acres you decided to work
int pplFood;//People food
int perAcre=3;//Bushels per acre
int ttlGrw[ENDYR]={0,0,0,0,0,0,0,0,0,0,0};//Array for grown each year
int ttlFed[ENDYR]={0,0,0,0,0,0,0,0,0,0,0};//The total grain for people/yr
int cnt=0;//Array increment counter
int deadYr[2][ENDYR]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};//dead/yr
vector<int> rats(10);//For adding up eaten by rats


//Display the first years data
dspYear(year,newPpl,strvd,pop,acres,totBush,perAcre,ratFood,lndPrc);



//Starting Loop For Years 1-11
for(year+=1;year<ENDYR;++year){
   //Displaying the output questions
   cout<<endl<<"How Many Acres Do You Wish To Buy/Sell:   ";
   cin>>sellBuy;          //Acres for sale/purchase question


   while(sellBuy<0&&sellBuy<(acres*-1)||sellBuy>0&&sellBuy>totBush/lndPrc){
      cout<<"\nI Am Afraid That Amount Is Not Possible Hammurabi-Senpai\n"
            "Enter Again\n";
   cin>>sellBuy;
}       //Validates for both selling more land than you have and for
        //purchasing more land than you could afford




//Updating and displaying important values
acres+=sellBuy;
totBush-=sellBuy*lndPrc; //Printing new values to help make decisions
cout<<endl<<"New Acres:   "<<acres<<endl;
cout<<"Total Bushels:     "<<totBush<<endl;

//Next question
cout<<"How Many Grains Would You Like To Feed Your People:   ";
cin>>pplFood;       //Feeding people question & answer
while(pplFood<=0||pplFood>totBush){
   cout<<"\nThat Is Not A Possible Amount My King\n"
           "Enter Feeding Grains\n";
   cin>>pplFood;       //Answer validation
}
//Fill Total fed to people array
ttlFed[year-2]=pplFood;

//Decision making info output
totBush-=pplFood;
strvd=loss(pop,pplFood);
cout<<"\nTotal Acres:       "<<acres;
cout<<"\nTotal Bushels:     "<<totBush<<endl;
cout<<"Total Population:  "<<pop<<endl;

deadYr[0][year-2]=year-1;//Setting years of array inside main for loop
deadYr[1][year-2]=strvd;//Setting starved of array inside main for loop

//Question 3
cout<<"How Many Acres Do You Wish To Plant With Seed:   ";
cin>>acrsWrk;       //Crop growing answer
while(acrsWrk>acres||acrsWrk<0||acrsWrk>pop*100){
   cout<<"\nThat Is Not Possible, My Lord\n"
           "Enter Again\n";
   cin>>acrsWrk;       //Answer validation
   cout<<endl;
}
//Updating total bushels for next calculations
totBush-=acrsWrk;
cout<<endl<<endl<<endl<<endl;
```

```
    //Next Year Data Processing
    pop-=strvd;//Taking away the people who died

    //Game Failure                    //If number dead exceeds 45% in 1
    if(strvd>=static_cast<float>(pop)*0.45){//year the game ends and loses
        year=12;//Ending for loop
    }
    //Continue Data Processing For Next Iteration
    lndPrc=priceL();//Random land price
    newPpl=neoPop();//Random new population
    pop+=newPpl;//Adding new population to old population
    perAcre=cropRnd();//Crops grown per acre
    crops=perAcre*acrsWrk;//Total Acres worked
    ttlGrw[year-2]=crops;
    totBush+=crops;//Total bushels after getting crops
    ratFood=loss(totBush);//Eaten by rats. takes away from totBush
    totBush-=ratFood;//Supply minus amount eaten by rats
    rats[year-2]=ratFood;//Rat total vector




    int plague=rand()%6;        //Plague randomizer
    if(plague==3){              //If plague happens
        pop=pop*(rand()%45 +51);
        pop/=100;
        cout<<"\nOh no, a violent plague occurred and killed many "
                "citizens\n";//Alerting player of tragedy
    }

      //Displaying recurring header for each year
dspYear(year,newPpl,strvd,pop,acres,totBush,perAcre,ratFood,lndPrc);

}

//Output for losing with no land
if(year==11&&acres<1){
    cout<<endl<<endl<<endl<<"You Are A King Without A Kingdom And A Failure"
            " Of A Man\nYou Finished With No Land And "<<pop<<" People\n"
            "\nYou Lose";
}          //Congratulations output
else if(year==11&&acres>0){
    cout<<"\n\n\nCongratulations, You Survived "
            "All Ten Years Without Failing\n"
            "Horribly And Causing Mass Genocides And Revolts\n";
        cout<<"You Ended With "<<pop<<" People And "<<acres<<" Acres.\n"
            "That Averages To About "<<acres/static_cast<float>(pop)
            <<" Acres Per Person\n\nYou Win";
    (acres/static_cast<float>(pop)>=12)?cout<<"\nYou Lead A Country Like Trump":
        cout<<"\nYou Have Not Yet Reached Trump Status";
    }   //Compares you with great leaders based on score like original does

    else{//Killing too many people output
        cout<<"\nYou Have Killed "<<strvd<<" people in 1 year\n"
                "You Have Been Dethroned And Executed For Incompetence\n"
                "You Lose The Game\n";
    }
    stats(deadYr,ttlFed,ttlGrw,rats);//Calling stats function after game ends

    //Asking to play again or quit
    cout<<"\n\n\n1.) Play Again\n2.) Quit\n";
    cin>>plyAgn;        //play again answers
    if(plyAgn>2||plyAgn<1){
        cout<<"\nInvalid Answer Input\nEnterAgain\n";
        cin>>plyAgn;//Answer validation
        if(plyAgn>9){//Nested loop
            cout<<"Enter A Single Digit Number From 1 To 2 To Proceed\n";
            cin>>plyAgn;
        }
    }
    switch(plyAgn){
        case 1:cout<<"\nGame Restarting\n\n\n";break;//Playing Again option
        case 2:cout<<"\nGame Ending\n\n\n";         //Ending Game option
    }

    }
    while(plyAgn!=2);//Replays game if choice does not equal two
    //A Sebastian Production
    return 0;
}


void gtTitle(){
    //Opening and writing to the Rules File
    ofstream title;     //Input file variable name
    string ttl;         //Variable used to read file info to

    title.open("title.dat");        //Opening file

    title<<"Hammurabi: A Game Of Strategy";//Reading title to file
    title.close();          //Closing file

    ifstream titleO;        //Input file variable
    titleO.open("title.dat");       //Opening input file

    while(titleO>>ttl){         //Displaying file name one string at a time
        cout<<ttl<<" ";
    }
    titleO.close();//Closing file
    cout<<endl;
}

void seeRule(){
    char ans;               //The answer given (just to check off char)
    bool choice;            //Boolean value for the rules display choice

    cout<<"Press 0 And Enter To See The Game Rules.\nPress "//Rules input prompt
            "1 To Continue And Play The Game\n";
    cin>>ans;       //inputting choice to see rules
    choice=ans-48;   //Setting the char to the boolean

    if(choice==false){
        cout<<"You Are Hammurabi. Ruler Of This Land\n\n"
    "1.) The game will last 10 rounds each being one year\n"
    "2.) Each living person needs 20 bushels of grain per\n"//The Game Rules
    <<setw(4)<<""<<"year and can work up to 10 acres of land annually\n"
    "3.) Each acre of land requires 1 bushel to farm on it\n"
    "4.) If you kill enough people in one year you will be\n"
    <<setw(4)<<""<<"impeached and lose the game\n"
    "5.) Enter a negative value to sell land, positive to buy\n"
    "6.) Reach year 11 successfully to win the game\n\n";

        cout<<"Press Enter To Play\n";
        cin.ignore();       //Clear null terminator out of keyboard buffer
        cin.get();          //Enter to go to the next screen
    }
    cout<<endl<<endl;
}

void dspYear(int &year,float newPpl,int strvd,float &pop,int &acres
,int &totBush,int perAcre,int ratFood, int lndPrc=23){
    cout<<setprecision(0)<<fixed;           //Making all outputs wholenumbers
    cout<<"Hammurabi: I beg to report to you,\n"
            "In year "<<year<<endl;                 //Header similar to real
    cout<<strvd<<" People starved\n";               //In game header
    cout<<newPpl<<" People came to the city\n";
    cout<<"The city population is now "<<pop<<endl;
    cout<<"The city now owns "<<acres<<" acres\n";
    cout<<"You harvested "<<perAcre<<" bushels per acre\n";
    cout<<"Rats ate "<<ratFood<<" bushels\n";
    cout<<"You now have "<<totBush<<" grains in store\n";
    cout<<"Land is trading at "<<lndPrc<<" bushels per acre\n";

}

short priceL(){
    short lndPrc=rand()%10+17;//Assigning land price to random

    return pow(lndPrc,1);//Returning land price for each round
//Just to technically use cmath . No uses of it in this program
}

short neoPop(){
    //Calculating new people each year
    short newPop=rand()%10+3;//Range [3,17]
    return newPop;//returning the amount of new people
}

short cropRnd(){
    //Calculating random crop variable each year
    short perAcre=0;//Initialize to 0
    perAcre=rand()%5+1;//Range [1,5] crops per acre

    return perAcre;//Returning bushels per acre variable
}

short loss(int totBush){
    //Calculating random possibility of rats
    short poss=0;//Start at 0
    poss=rand()%3+1;//Possibility of rats eating grain is 1/3
    short eaten=0;//The numeric amount eaten

    if(poss==1){//If poss =1 then rats will come else
    float perc;//Percentage of crops ravaged by rats
    perc=rand()%50+1;//Range of [1,50] percent
    eaten=(totBush*perc/100);
    }
    else
        eaten=0;//No rats = np food eaten
    return eaten;//send back amount eaten
}
```

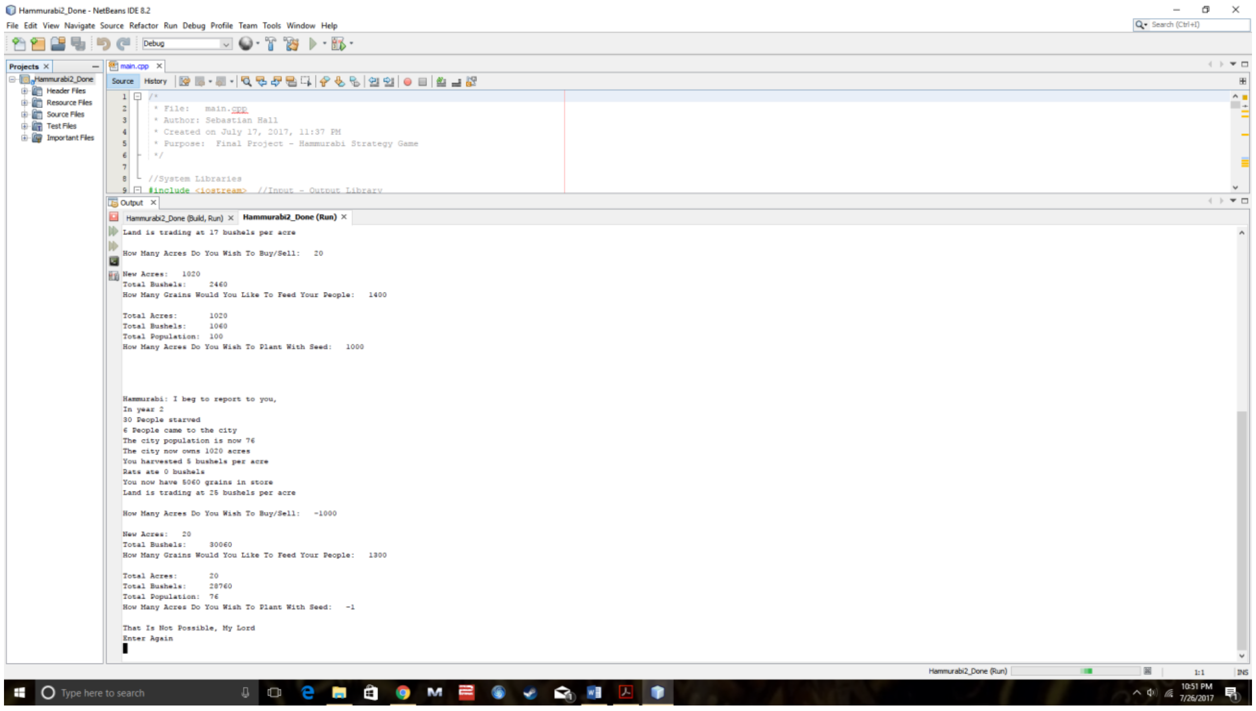
```
int loss(float pop,int pplFood){
    short strvd;
    if(pplFood/20<=pop){
        strvd=pop-pplFood/20;//Calculate how many starved if inadequate food
    }                         //Is offered
    else
        strvd=0;    //If enough food is given, strvd is default 0
    return strvd;//Return dead people
    }


void stats(int deadYr[][11],int ttlFed [],int ttlGrw[],vector<int> rats){
    cout<<"\n\nYear: ";
    int tGrw=0;//Total grown
    int tFed=0;//Total fed
    int highest=0;
    int ratTtl=0;

    for(int i=0;i<2;i++){
        for(int j=0;j<10;j++){
            cout<<setw(2)<<deadYr[i][j]<<" ";
        }
        if(i==0);
        cout<<endl<<"Dead: ";
    }
    //Sorting deadYr
    bool swap;
    int temp;
    do{
        swap=false;
        for(int i=0;i<2;i++){
            for(int j=0;j<ENDYR;j++){
                if(deadYr[i]>deadYr[i+1]){
                    temp=deadYr[i][j];            //Sorting before searching
                    deadYr[i][j]=deadYr[i][j+1];
                    deadYr[i][j+1]=temp;
                    swap=true;
                }
            }
        }
    }while(swap);

    //Calculate totals from arrays
    for(int i=0;i<ENDYR-1;i++){
        tFed+=ttlFed[i];
    }
    //Calculate totals from arrays
    for(int i=0;i<ENDYR-1;i++){
        tGrw+=ttlGrw[i];
    }
    //Calculate totals from arrays
    for(int i=0;i<ENDYR-1;i++){
        ratTtl+=rats[i];
    }
    //Getting high from linear search
    for(int i=0;i<ENDYR-1;i++){
        if(deadYr[1][i]>highest)
            highest=deadYr[1][i];
    }
    //Output remaining statistics
    cout<<endl<<"Highest Killed In One Year: "<<highest<<endl<<
            "Total Grains Fed: "<<tFed<<endl<<"Total Grown: "<<tGrw<<endl
            <<"Total eaten by rats: "<<ratTtl<<endl;


}
```

Sample I/O



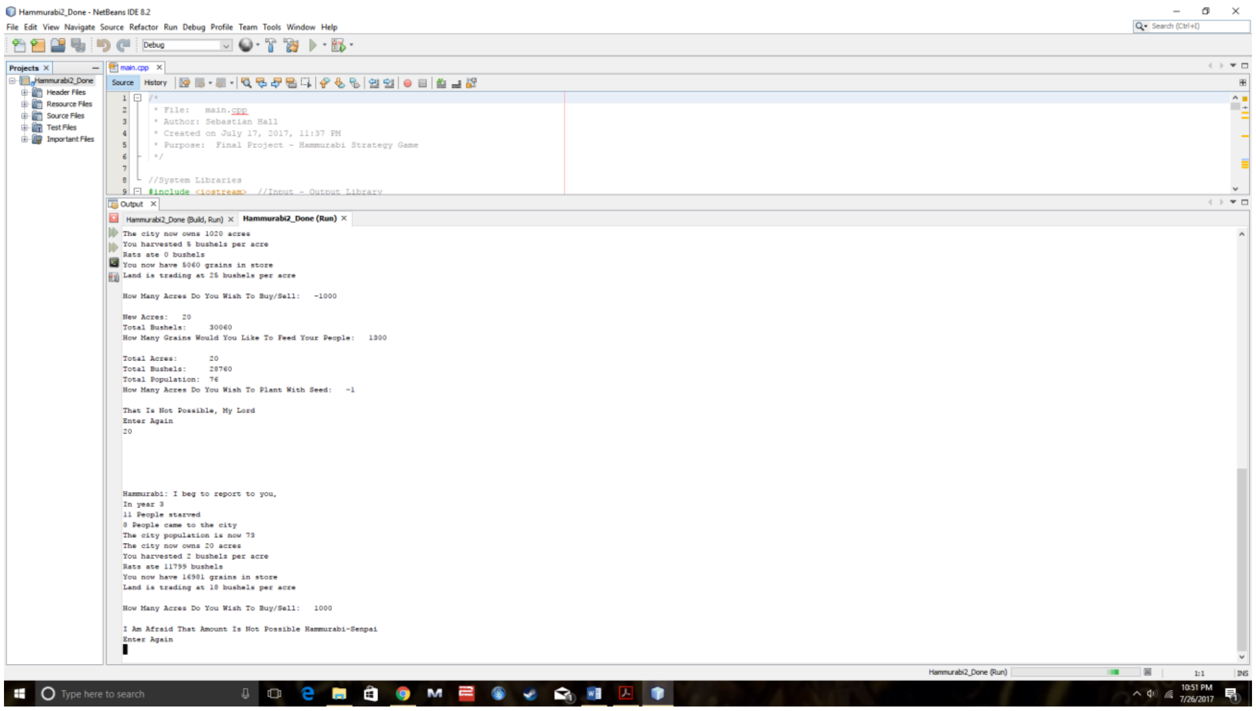
```
/*
 * File:   main.cpp
 * Author: Sebastian Hall
 * Created on July 17, 2017, 11:37 PM
 * Purpose:  Final Project - Hammurabi Strategy Game
 */

//System Libraries
#include <iostream>  //Input - Output Library
```

```
Hammurabi: A Game Of Strategy

Press 0 And Enter To See The Game Rules.
Press 1 To Continue And Play The Game
0
You Are Hammurabi. Ruler Of This Land

1.) The game will last 10 rounds each being one year
2.) Each living person needs 20 bushels of grain per
    year and can work up to 10 acres of land annually
3.) Each acre of land requires 1 bushel to farm on it
4.) If you kill enough people in one year you will be
    impeached and lose the game
5.) Enter a negative value to sell land, positive to buy
6.) Reach year 11 successfully to win the game

Press Enter To Play
```



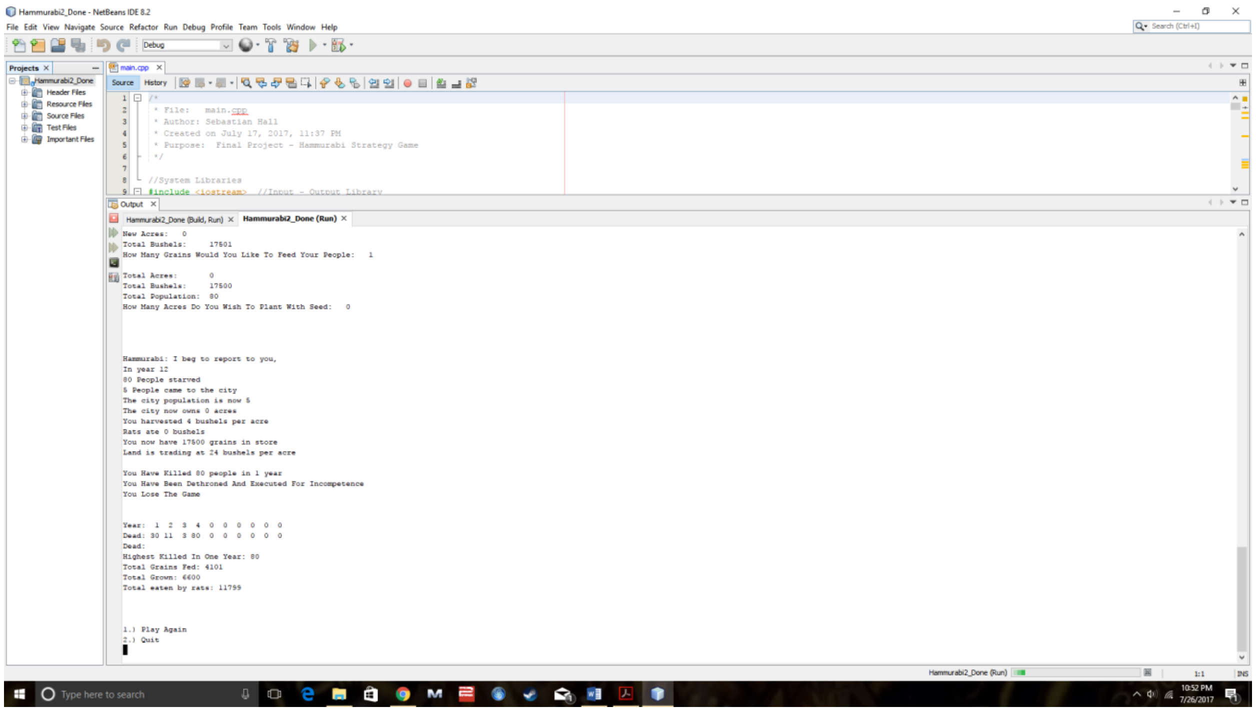
```
/*
 * File:   main.cpp
 * Author: Sebastian Hall
 * Created on July 17, 2017, 11:37 PM
 * Purpose:  Final Project - Hammurabi Strategy Game
 */

//System Libraries
#include <iostream>  //Input - Output Library
```

```
Land is trading at 17 bushels per acre

How Many Acres Do You Wish To Buy/Sell:   20

New Acres:   1020
Total Bushels:      2460
How Many Grains Would You Like To Feed Your People:   1400

Total Acres:        1020
Total Bushels:      1060
Total Population:  100
How Many Acres Do You Wish To Plant With Seed:   1000




Hammurabi: I beg to report to you,
In year 2
30 People starved
6 People came to the city
The city population is now 76
The city now owns 1020 acres
You harvested 5 bushels per acre
Rats ate 0 bushels
You now have 5060 grains in store
Land is trading at 25 bushels per acre

How Many Acres Do You Wish To Buy/Sell:   -1000

New Acres:   20
Total Bushels:      30060
How Many Grains Would You Like To Feed Your People:   1300

Total Acres:        20
Total Bushels:      28760
Total Population:  76
How Many Acres Do You Wish To Plant With Seed:   -1

That Is Not Possible, My Lord
Enter Again
```



```
/*
 * File:   main.cpp
 * Author: Sebastian Hall
 * Created on July 17, 2017, 11:37 PM
 * Purpose:  Final Project - Hammurabi Strategy Game
 */

//System Libraries
#include <iostream>  //Input - Output Library
```

```
The city now owns 1020 acres
You harvested 5 bushels per acre
Rats ate 0 bushels
You now have 5060 grains in store
Land is trading at 25 bushels per acre

How Many Acres Do You Wish To Buy/Sell:   -1000

New Acres:   20
Total Bushels:     30060
How Many Grains Would You Like To Feed Your People:   1300

Total Acres:       20
Total Bushels:     28760
Total Population:  76
How Many Acres Do You Wish To Plant With Seed:   -1

That Is Not Possible, My Lord
Enter Again
20




Hammurabi: I beg to report to you,
In year 3
11 People starved
8 People came to the city
The city population is now 73
The city now owns 20 acres
You harvested 2 bushels per acre
Rats ate 11799 bushels
You now have 16981 grains in store
Land is trading at 18 bushels per acre

How Many Acres Do You Wish To Buy/Sell:   1000

I Am Afraid That Amount Is Not Possible Hammurabi-Senpai
Enter Again
```



```
/*
 * File:   main.cpp
 * Author: Sebastian Hall
 * Created on July 17, 2017, 11:37 PM
 * Purpose:  Final Project - Hammurabi Strategy Game
 */

//System Libraries
#include <iostream>  //Input - Output Library
```

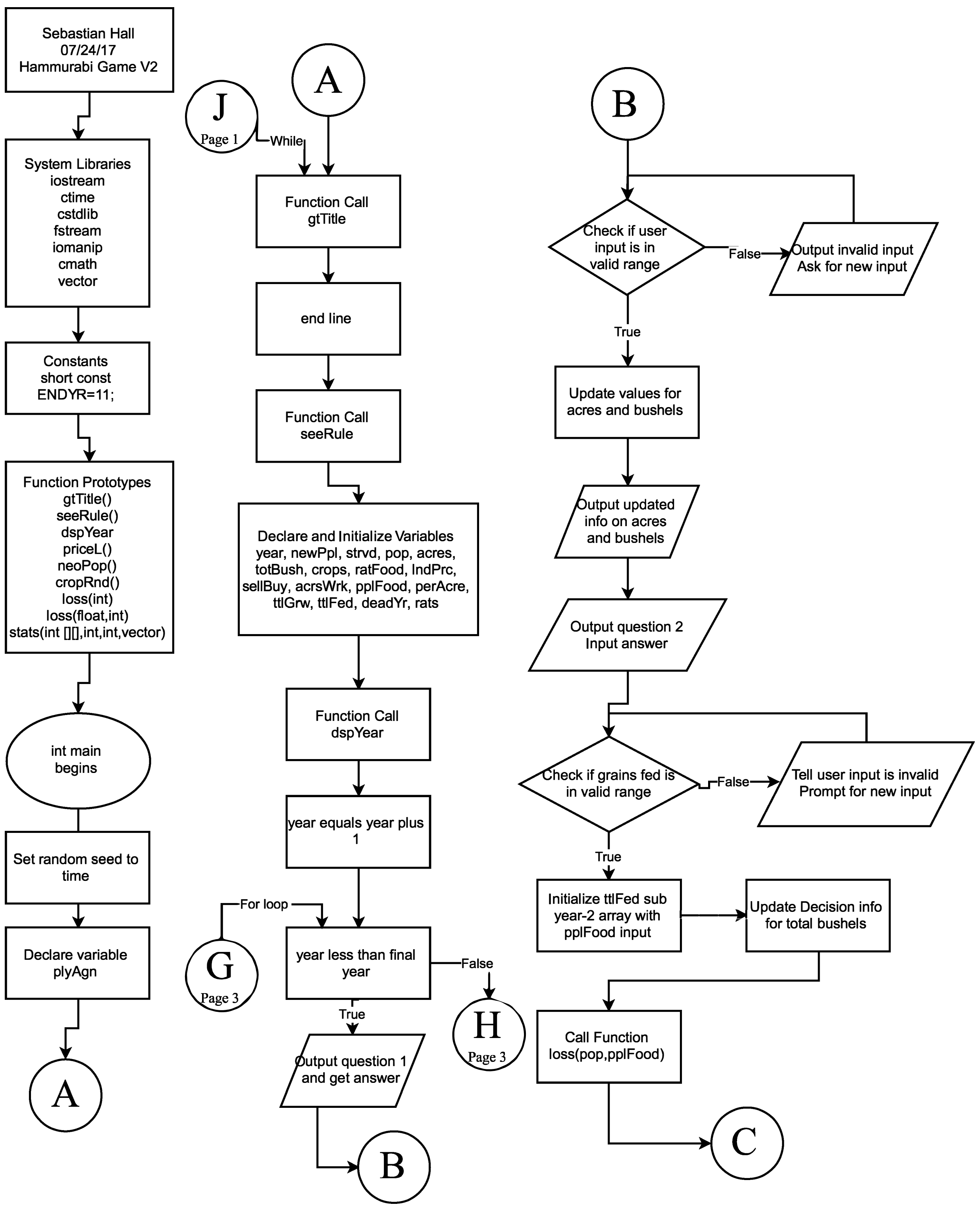
```
New Acres:   0
Total Bushels:     17501
How Many Grains Would You Like To Feed Your People:   1

Total Acres:       0
Total Bushels:     17500
Total Population:  80
How Many Acres Do You Wish To Plant With Seed:   0




Hammurabi: I beg to report to you,
In year 12
80 People starved
5 People came to the city
The city population is now 5
The city now owns 0 acres
You harvested 4 bushels per acre
Rats ate 0 bushels
You now have 17500 grains in store
Land is trading at 24 bushels per acre

You Have Killed 80 people in 1 year
You Have Been Dethroned And Executed For Incompetence
You Lose The Game


Year:  1  2  3  4  0  0  0  0  0  0
Dead: 30 11  3 80  0  0  0  0  0  0
Dead:
Highest Killed In One Year: 80
Total Grains Fed: 4101
Total Grown: 6600
Total eaten by rats: 11799



1.) Play Again
2.) Quit
```

Cross Reference for Project 2

Chapter	Section	Topic	Where in Code Line number
2	2	cout	Line 82
	3	libraries	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	Line 83 & Line 59
	5	Identifiers	Line 64
	6	Integers	Line 65
	7	Characters	Line 240
	8	Strings	Line 222 & 232
	9	Floats No Doubles	Line 55
	10	Bools	Line 241
	11	Sizeof *****	N/A
	12	Variables 7 characters or less	Line 53 - 71
	13	Scope ***** No Global Variables	N/A
	14	Arithmetic operators	Line 97
	15	Comments 20%+	All Throughout
	16	Named Constants	Line 21
	17	Programming Style ***** Emulate	N/A
3	1	cin	Line 104
	2	Math Expression	Line 161
	3	Mixing data types ****	N/A
	4	Overflow/Underflow ****	N/A
	5	Type Casting	Line 141
	6	Multiple assignment *****	N/A
	7	Formatting output	Line 345
	8	Strings	Line 222 & 232
	9	Math Library	Line 287
	10	Hand tracing ******	N/A
4	1	Relational Operators	Line 86
	2	if	Line 141
	4	If-else	Line 314 & Line 319
	5	Nesting	Line 335 & Line 336
	6	If-else-if	Line 171 & Line 178
	7	Flags *****	N/A
	8	Logical operators	Line 86
	11	Validating user input	Line 105
	13	Conditional Operator	Line 185
	14	Switch	Line 207
5	1	Increment/Decrement	Line 80
	2	While	Line 126
	5	Do-while	Line 44 & Line 213
	6	For loop	Line 376
	11	Files input/output both	Line 221 - Line 235
	12	No breaks in loops ******	N/A

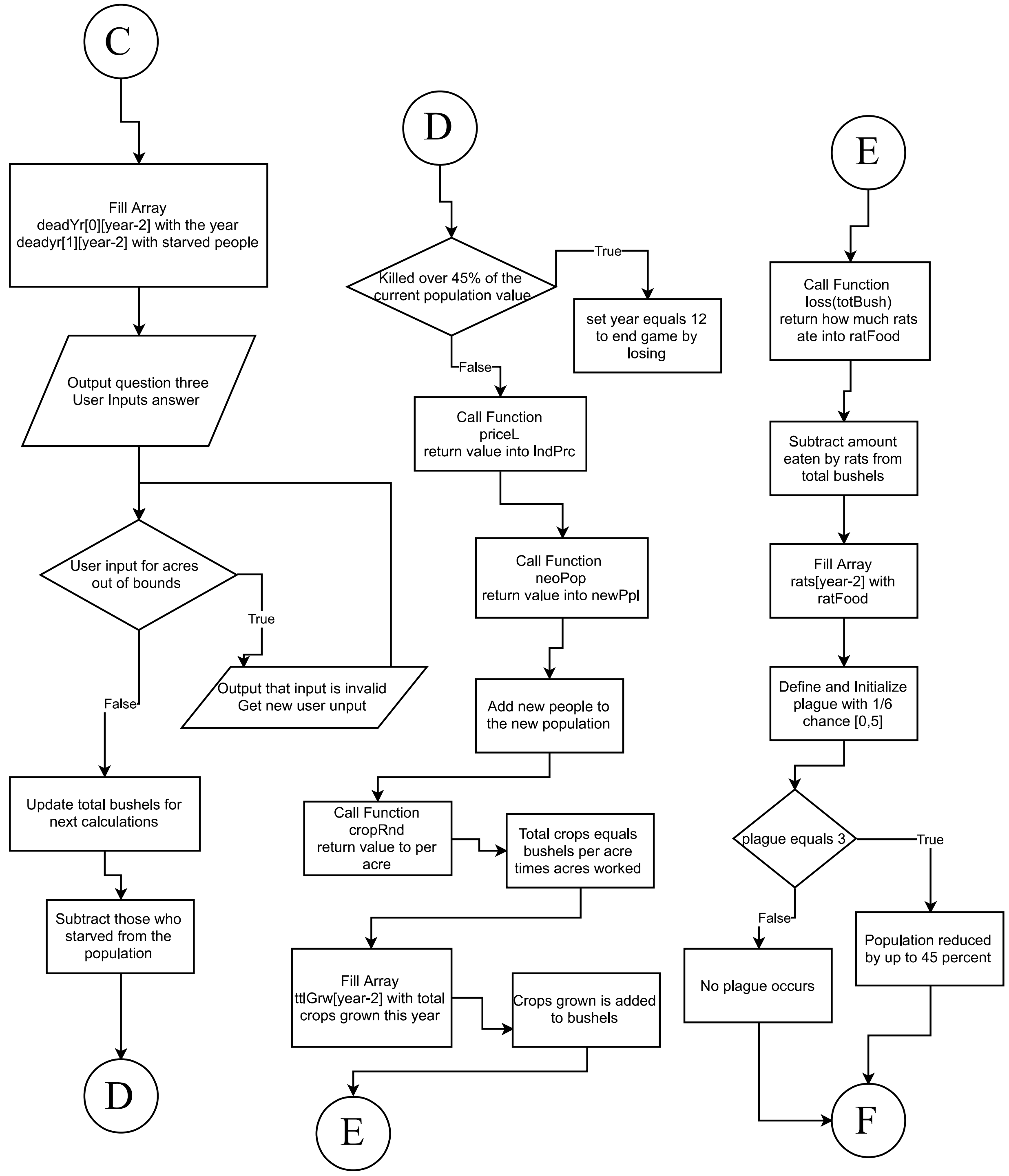
6	3	Function Prototypes	Line 24 - Line 33
	5	Passing by value	Line 168
	8	Returning values from functions	Line 287
	9	Returning a boolean ******	N/A
	10	No Global Variables Allowed	Line 21
		Only Global Constants	Line 168
		Meaning Conversions,Physical Con	Line 27
	11	Static Local	Line 54
	12	Default arguments	Line 267
	13	Reference Parameters	Line 266
	14	Overloading functions	Line 32 & Line 33 & Line 307 & Line 325
	15	Exit function *******	N/A
7	4	Array Initialization	Line 67
	6	Processing Arrays	Line 367 - Line 370
	7	Parallel Arrays	Line 67 & Line 87
	8	Arrays as function arguments	Line 194
	9	2-D Arrays	Line 70 & Line 343 - Line 349
	12	STL Vector	Line 71 & Line 154
8	1	Linear and Binary Search	Line 380 - Line 383
	3	Bubble and Selection Sort	Line 355 - Line 364
	5	Search/Sorting Vectors ******	N/A
****** Not required to show			

Flowchart

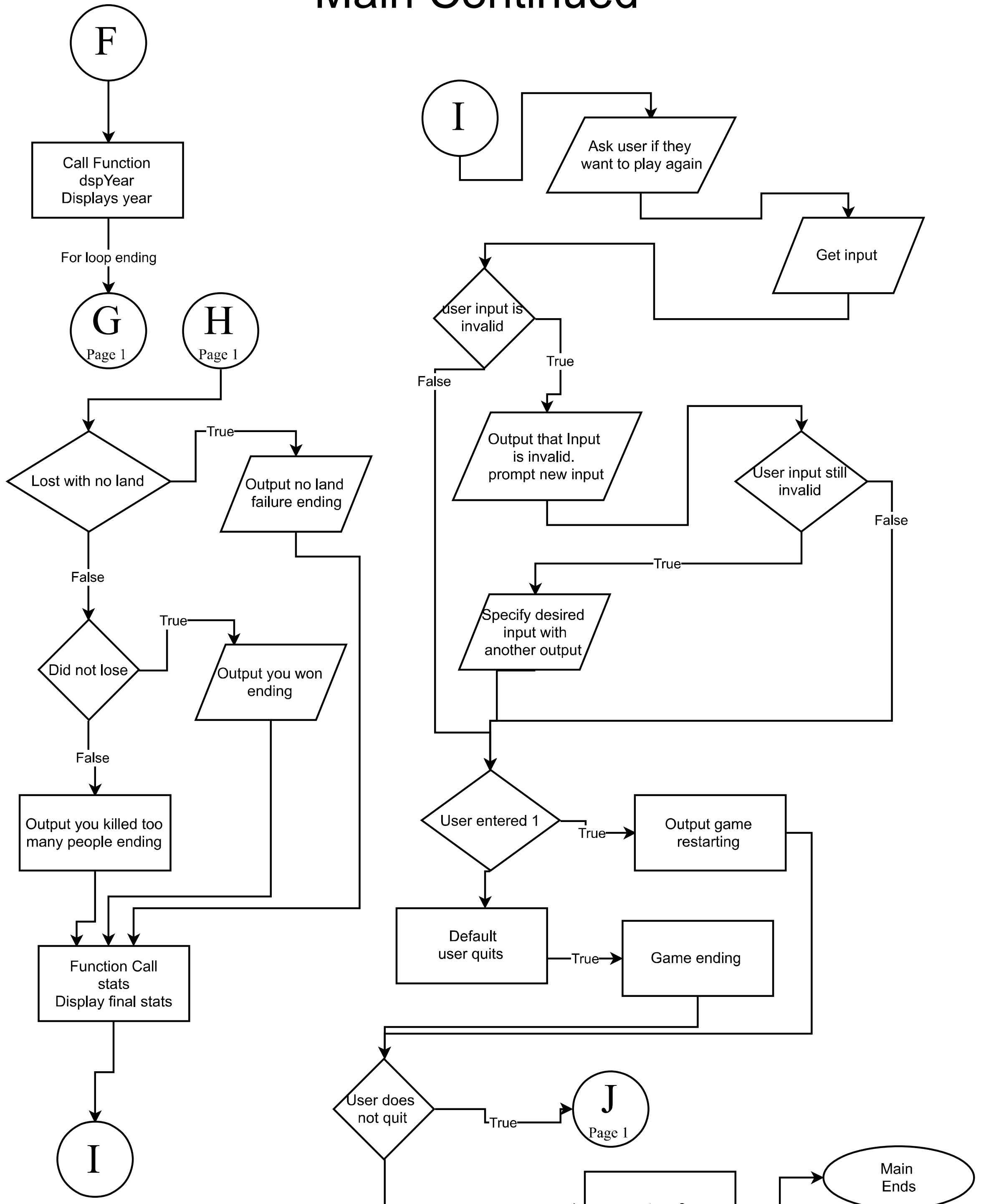


Sebastian Hall
07/24/17
Hammurabi Game V2
System Libraries
iostream
ctime
cstdlib
fstream
iomanip
cmath
vector
Constants
short const
ENDYR=11;
Function Prototypes
gtTitle()
seeRule()
dspYear
priceL()
neoPop()
cropRnd()
loss(int)
loss(float,int)
stats(int [][],int,int,vector)
int main
begins
Set random seed to time
Declare variable plyAgn
A
J
Page 1
While
A
Function Call
gtTitle
end line
Function Call
seeRule
Declare and Initialize Variables
year, newPpl, strvd, pop, acres, totBush, crops, ratFood, lndPrc, sellBuy, acrsWrk, pplFood, perAcre, ttlGrw, ttlFed, deadYr, rats
Function Call
dspYear
year equals year plus 1
For loop
G
Page 3
year less than final year
False
H
Page 3
True
Output question 1 and get answer
B
B
Check if user input is in valid range
False
Output invalid input
Ask for new input
True
Update values for acres and bushels
Output updated info on acres and bushels
Output question 2
Input answer
Check if grains fed is in valid range
False
Tell user input is invalid
Prompt for new input
True
Initialize ttlFed sub year-2 array with pplFood input
Update Decision info for total bushels
Call Function
loss(pop,pplFood)
C

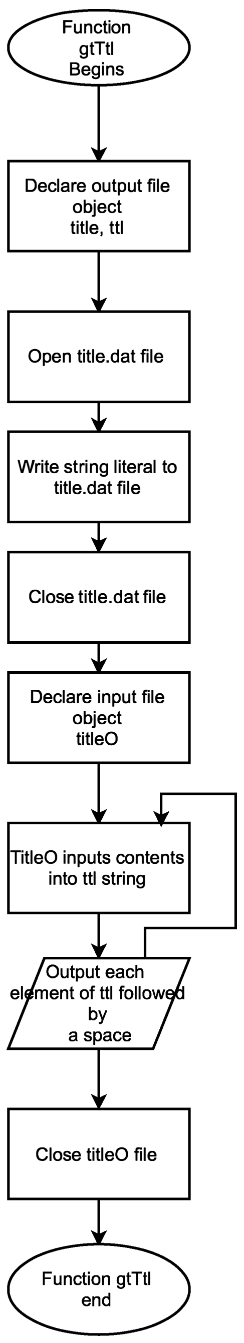
Main Continued



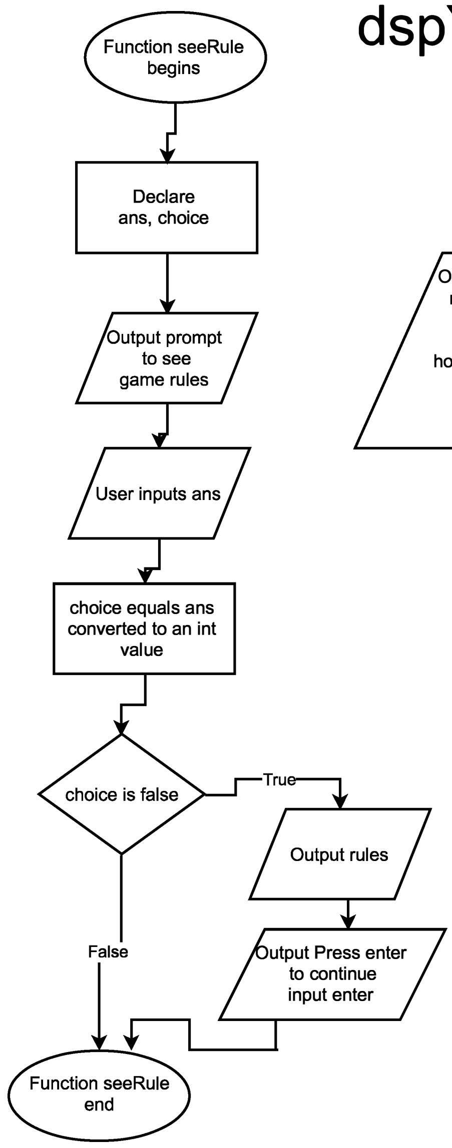
Main Continued



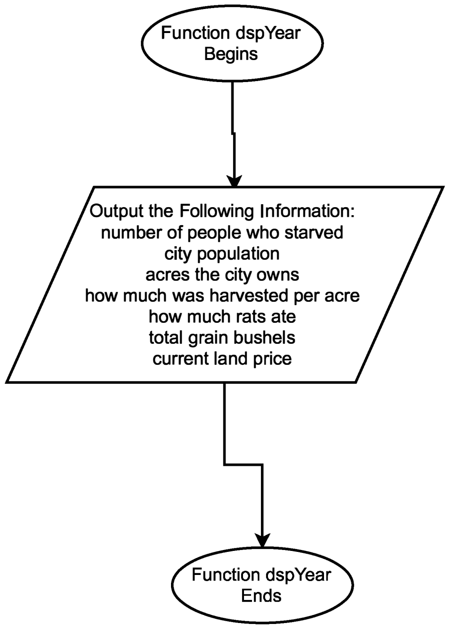
GtTtl Function


Function gtTtl Begins
Declare output file object title, ttl
Open title.dat file
Write string literal to title.dat file
Close title.dat file
Declare input file object titleO
TitleO inputs contents into ttl string
Output each element of ttl followed by a space
Close titleO file
Function gtTtl end

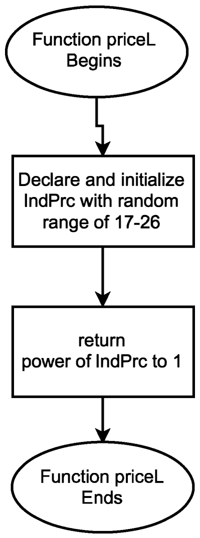
seeRuleFunction


Function seeRule begins
Declare ans, choice
Output prompt to see game rules
User inputs ans
choice equals ans converted to an int value
choice is false
True
Output rules
Output Press enter to continue input enter
False
Function seeRule end

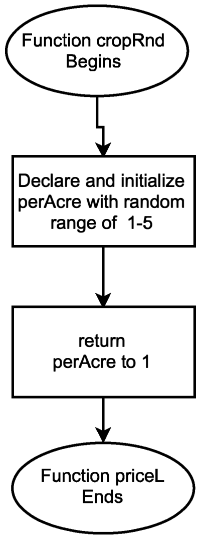
dspYear Function


Function dspYear Begins
Output the Following Information: number of people who starved city population acres the city owns how much was harvested per acre how much rats ate total grain bushels current land price
Function dspYear Ends

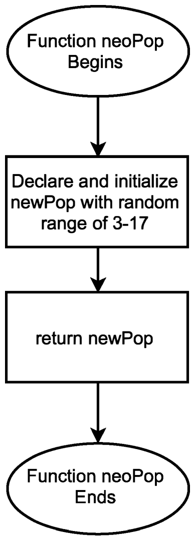
priceL Function


Function priceL Begins
Declare and initialize IndPrc with random range of 17-26
return power of IndPrc to 1
Function priceL Ends

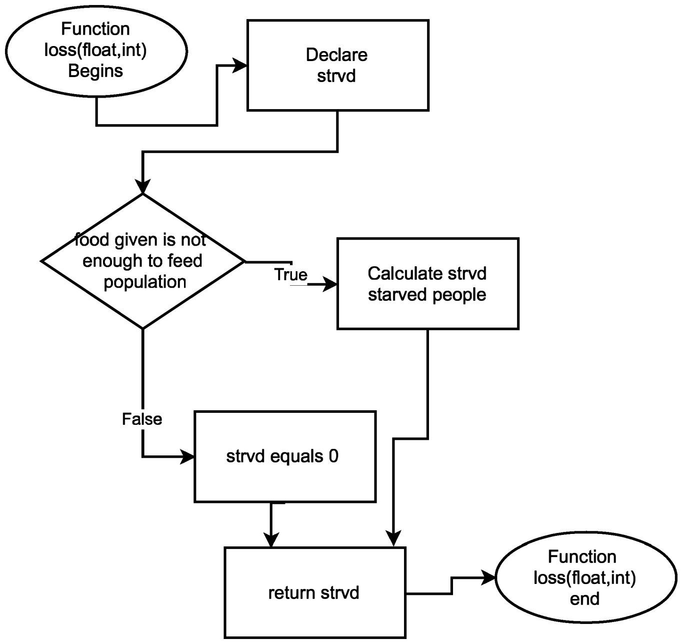
priceL Function



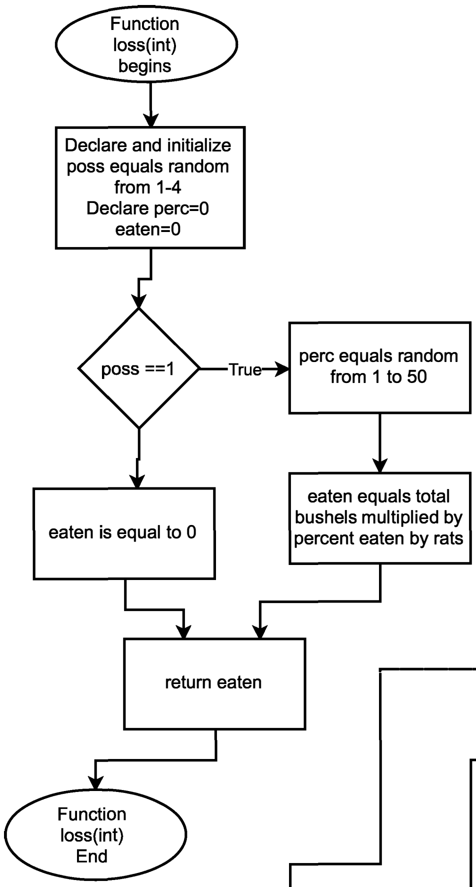
neoPop Function



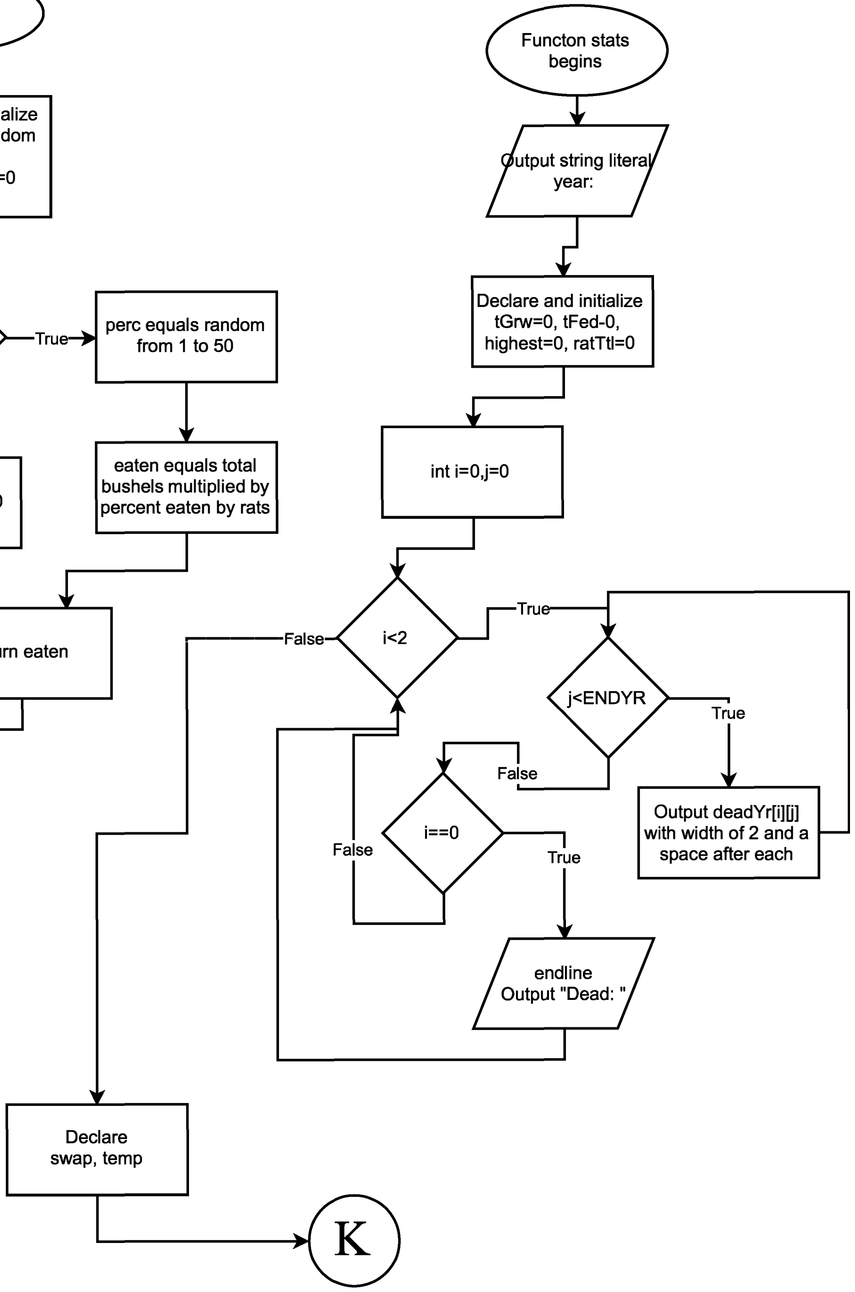
loss(float,int) Function


Function loss(float,int) Begins
Declare strvd
food given is not enough to feed population
True
Calculate strvd starved people
False
strvd equals 0
return strvd
Function loss(float,int) end

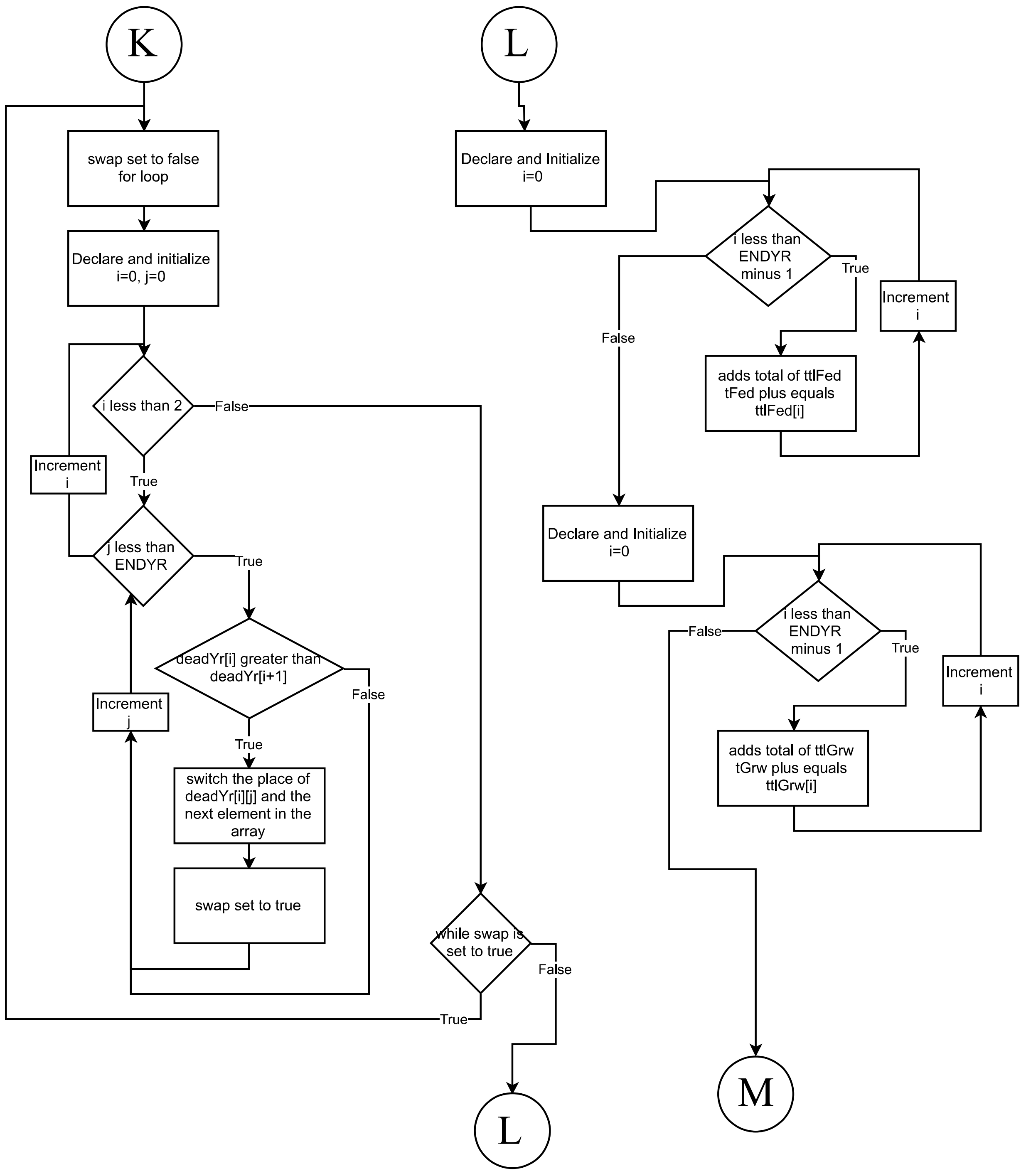
priceL Function


Function loss(int) begins
Declare and initialize poss equals random from 1-4 Declare perc=0 eaten=0
poss ==1
True
perc equals random from 1 to 50
eaten is equal to 0
eaten equals total bushels multiplied by percent eaten by rats
return eaten
Function loss(int) End

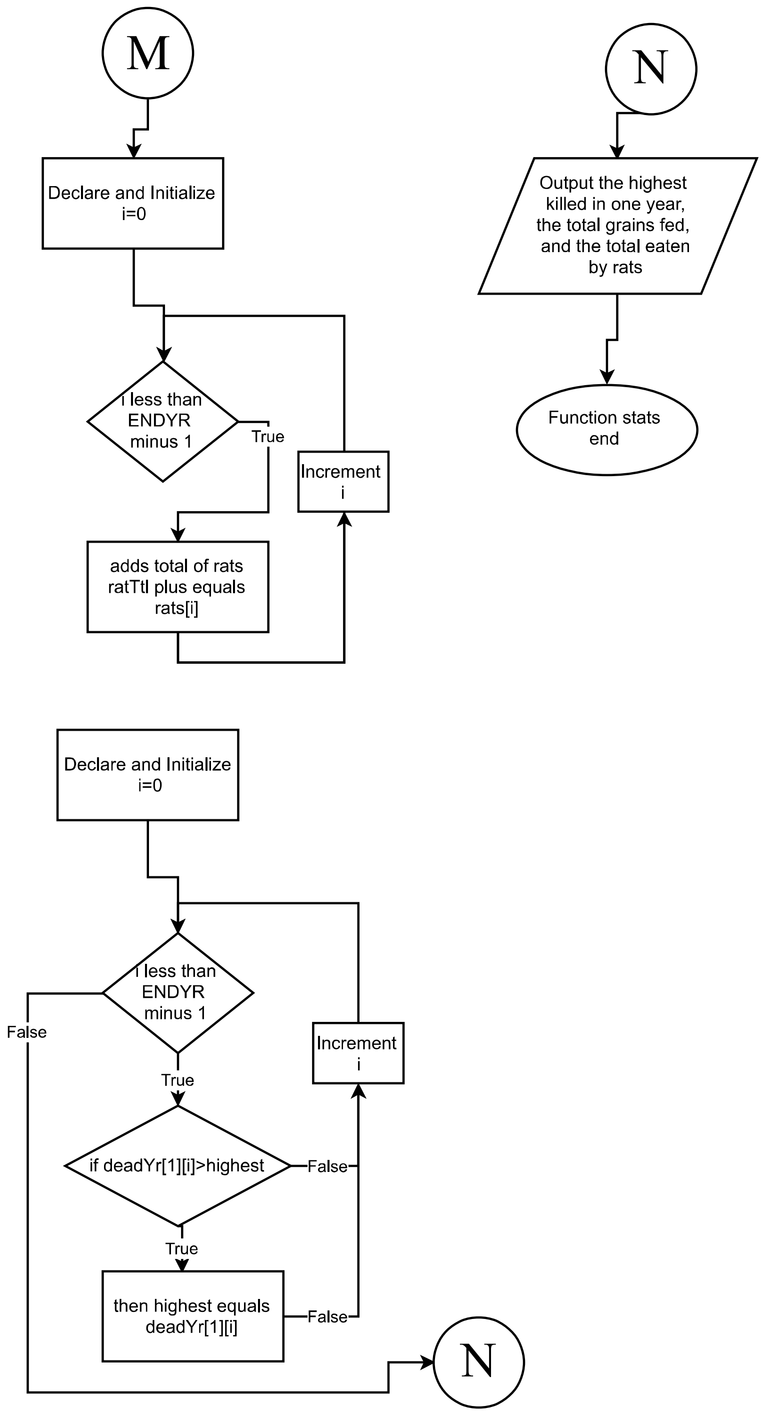
stats Function


Functon stats begins
Output string literal year:
Declare and initialize tGrw=0, tFed-0, highest=0, ratTtl=0
int i=0,j=0
i<2
True
False
j<ENDYR
True
False
Output deadYr[i][j] with width of 2 and a space after each
i==0
True
False
endline Output "Dead: "
Declare swap, temp
K

stats Function Continued


K
swap set to false for loop
Declare and initialize i=0, j=0
i less than 2
False
True
Increment i
j less than ENDYR
True
deadYr[i] greater than deadYr[i+1]
False
True
switch the place of deadYr[i][j] and the next element in the array
swap set to true
Increment j
while swap is set to true
False
True
L
L
Declare and Initialize i=0
i less than ENDYR minus 1
True
False
Increment i
adds total of ttlFed tFed plus equals ttlFed[i]
Declare and Initialize i=0
i less than ENDYR minus 1
False
True
Increment i
adds total of ttlGrw tGrw plus equals ttlGrw[i]
M

stats Function Continued


M
Declare and Initialize
i=0
i less than
ENDYR
minus 1
True
Increment
i
adds total of rats
ratTtl plus equals
rats[i]
N
Output the highest
killed in one year,
the total grains fed,
and the total eaten
by rats
Function stats
end
Declare and Initialize
i=0
i less than
ENDYR
minus 1
False
True
Increment
i
if deadYr[1][i]>highest
False
True
then highest equals
deadYr[1][i]
False
N