Hammurabi: Table Of Contents

Sebastian Hall
07/21/17

Hammurabi: Text Game

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
    Update and output decision making info
    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 rat food 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
    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
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
using namespace std; //Name-space under which system libraries exist

//User Libraries

//Global Constants

//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 ratFd(int);//Percentage eaten by rats
int india(float ,int);



//Execution begins here
int main() {
   //Setting random seed
   srand(static_cast<unsigned int>(time(0)));
   //Constants
   short const ENDYR=11;//The year on which all games of Hammurabi end

   //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
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

//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<11;++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
}
//Decision making info output
totBush-=pplFood;
strvd=india(pop,pplFood);
cout<<"\nTotal Acres:      "<<acres;
cout<<"\nTotal Bushels:     "<<totBush<<endl;
cout<<"Total Population:  "<<pop<<endl;

//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
   cout<<"You Have Killed "<<strvd<<" people in 1 year\n"
         "You Have Been Dethroned And Executed For Incompetence\n"
         "You Lose The Game\n";break;//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
totBush+=crops;//Total bushels after getting crops
ratFood=ratFd(totBush);//Eaten by rats. takes away from totBush
totBush-=ratFood;
```

```
    //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


//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();
    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
          "Anything Else 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){

    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<<" in store\n";
    cout<<"Land is trading at "<<lndPrc<<" bushels per acre\n";

}

short priceL(){
    //Setting the random seed
    srand(static_cast<unsigned int>(time(0)));
    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(){
   //Setting the random seed
   srand(static_cast<unsigned int>(time(0)));
   short newPop=rand()%10+3;//Range [3,17]
   return newPop;//returning the amount of new people
}

short cropRnd(){
   //Random Seed Set
   srand(static_cast<unsigned int>(time(0)));
   short perAcre=0;//Initialize to 0
   perAcre=rand()%5+1;//Range [1,5] crops per acre

   return perAcre;//Returning bushels per acre variable
}

short ratFd(int totBush){
   //Time seed random
   srand(static_cast<unsigned int>(time(0)));
   short poss=rand()%3+1;//Possibility of rats eating grain is 1/3
   short eaten=0;//The numeric amount eaten

   if(poss==2){//If poss =2 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 india(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

    }
```

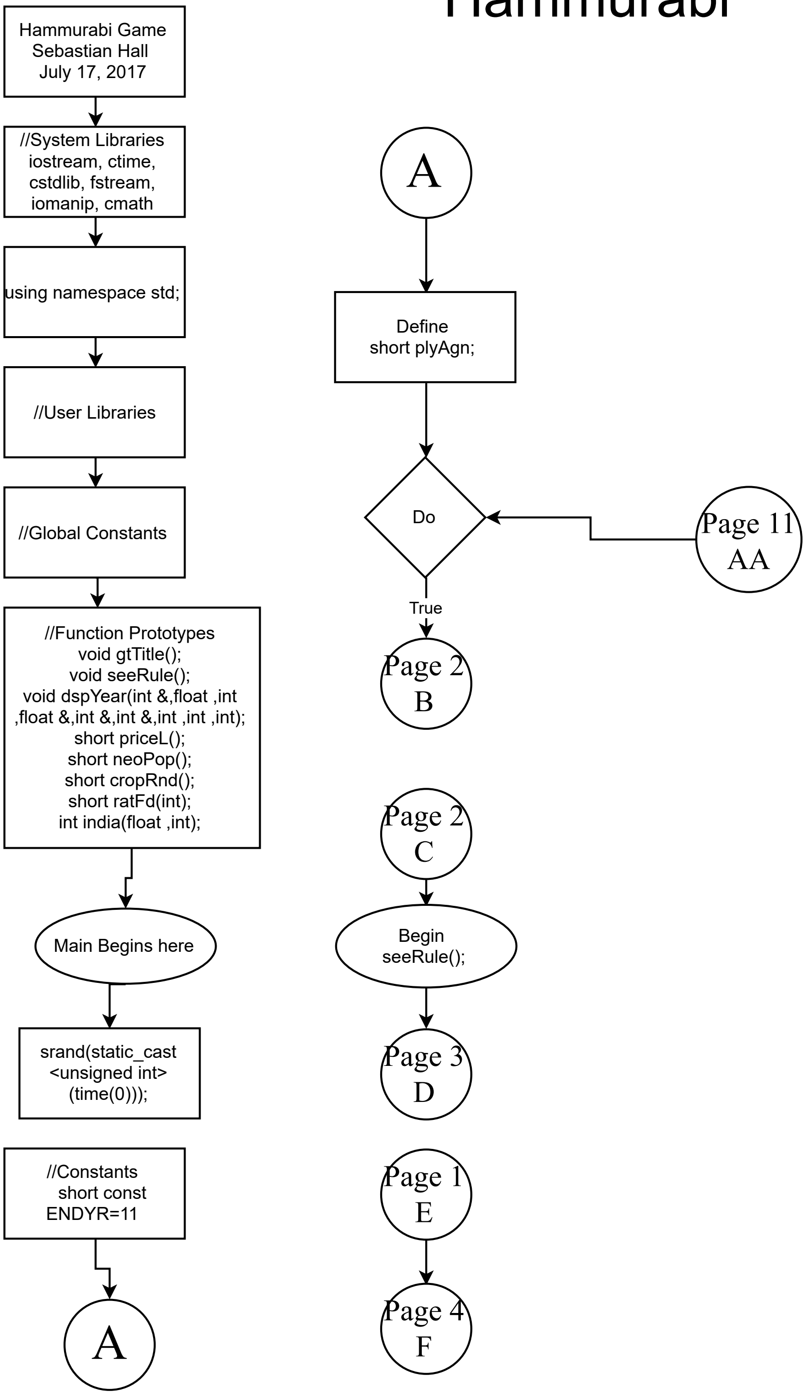
End Program

Cross Reference for Project 1

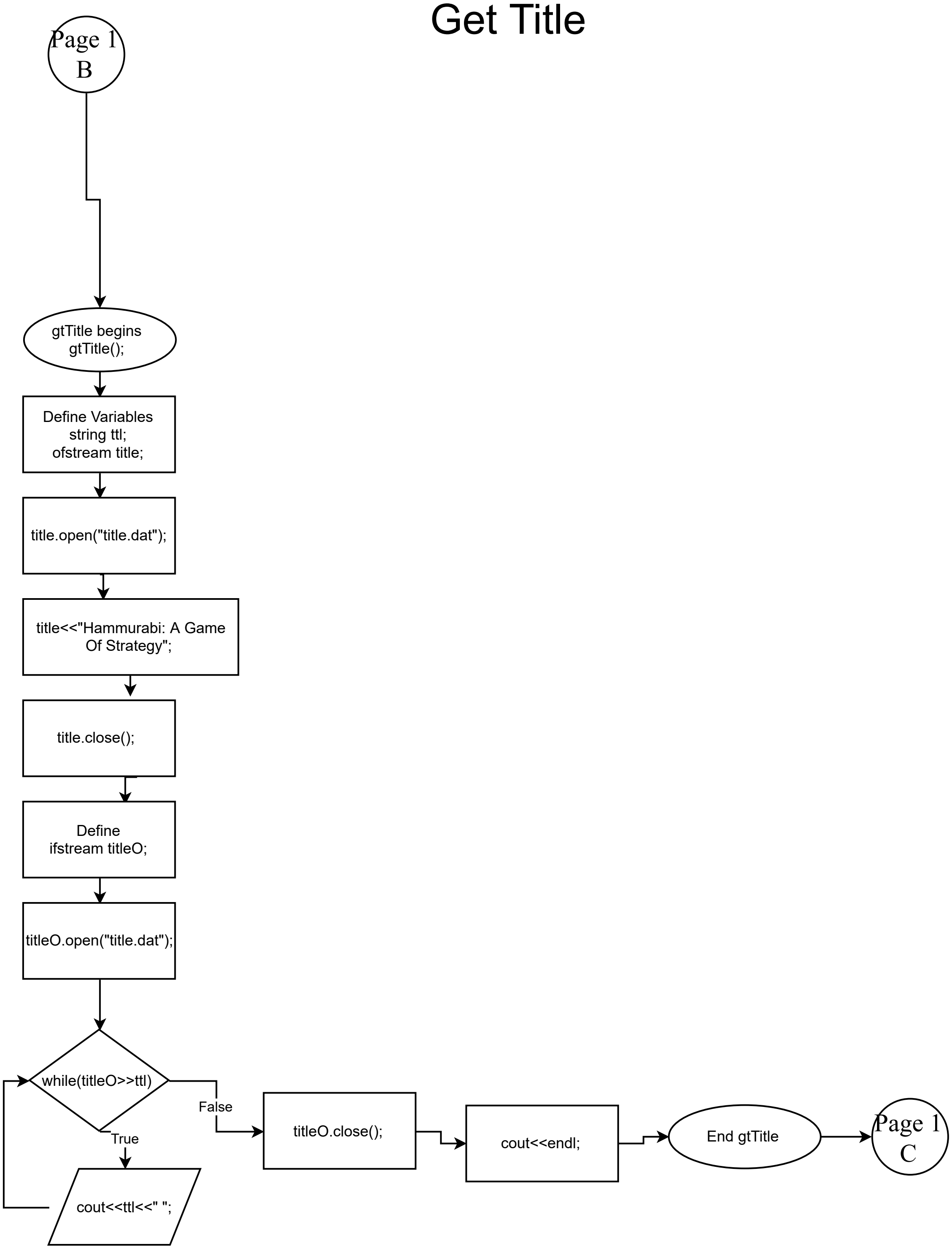
Column1	Column2	Column3	Where in Code
Chapter	**Section**	**Topic**	**Line number**
2	2	cout	Line 80
	3	libraries	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	Line 54
	5	Identifiers	Line 47-66
	6	Integers	Line 58
	7	Characters	Line 217
	8	Strings	Line 200 & 211
	9	Floats No Doubles	Line 55
	10	Bools	Line 218
	11	Sizeof *****	N/A
	12	Variables 7 characters or less	Lines 54-66
	13	Scope ***** No Global Variables	N/A
	14	Arithmetic operators	Line 286-293
	15	Comments 20%+	All over the place
	16	Named Constants	Line 38
	17	Programming Style ***** Emulate	N/A
3	1	cin	Line 81
	2	Math Expression	Line 96
	3	Mixing data types ****	N/A
	4	Overflow/Underflow ****	N/A
	5	Type Casting	Line 134
	6	Multiple assignment *****	N/A
	7	Formatting output	Line 229 & 232
	8	Strings	Line 211
	9	Math Library	Line 14 & 264
	10	Hand tracing ******	N/A
4	1	Relational Operators	Line 84
	2	if	Line 134
	4	If-else	Line 290 & 295
	5	Nesting	Line 176-183
	6	If-else-if	Line 157 & 162
	7	Flags *****	N/A
	8	Logical operators	Line 84
	11	Validating user input	Line 84-88
	13	Conditional Operator	Line 169
	14	Switch	Line 184
5	1	Increment/Decrement	Line 77
	2	While	Line 104
	5	Do-while	Line 44 & 191
	6	For loop	Line 77
	11	Files input/output both	Line 197-215
	12	No breaks in loops ******	N/A

****** Not required to show			

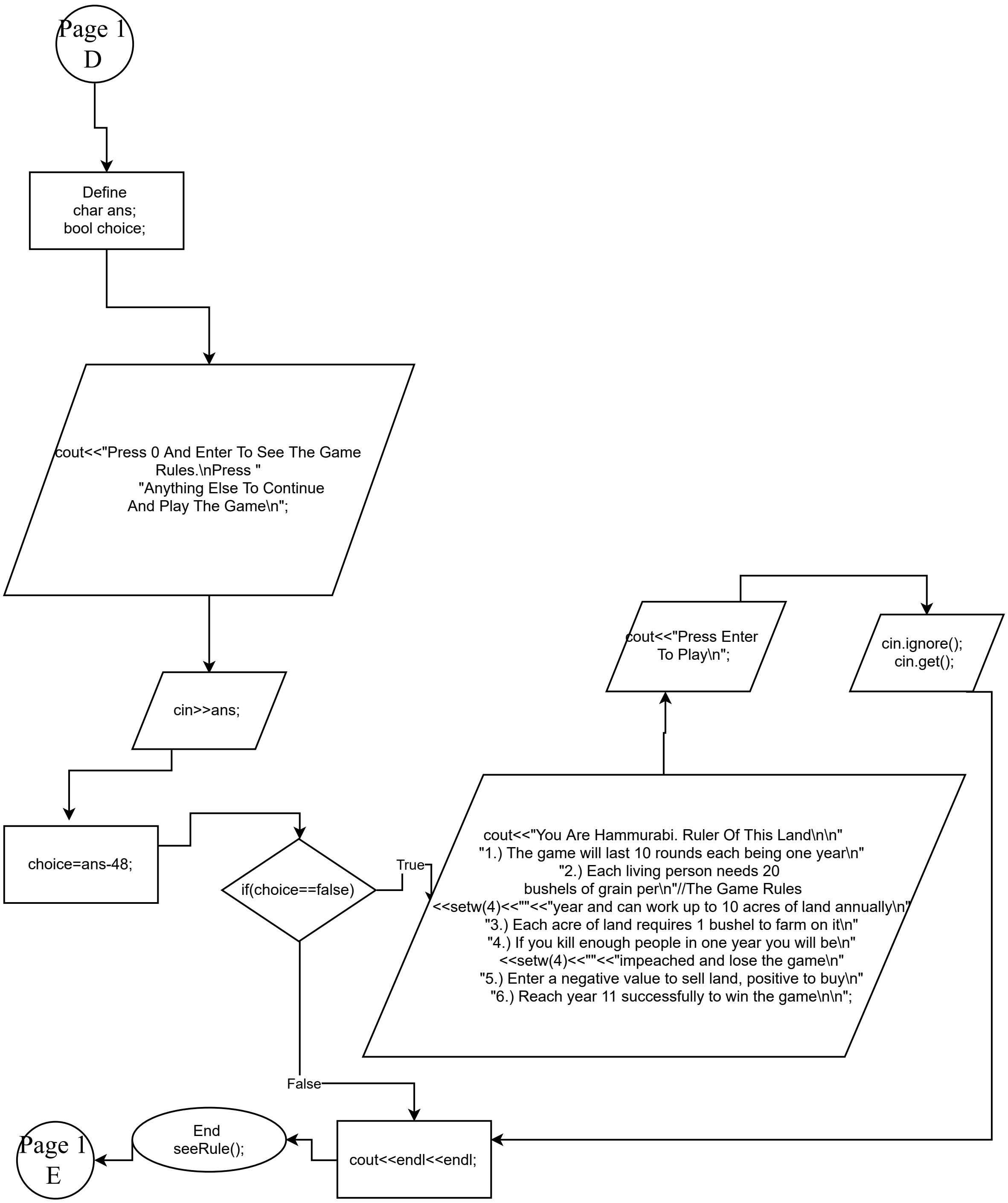
Hammurabi


Hammurabi Game
Sebastian Hall
July 17, 2017
//System Libraries
iostream, ctime,
cstdlib, fstream,
iomanip, cmath
using namespace std;
//User Libraries
//Global Constants
//Function Prototypes
void gtTitle();
void seeRule();
void dspYear(int &,float ,int
,float &,int &,int &,int ,int ,int);
short priceL();
short neoPop();
short cropRnd();
short ratFd(int);
int india(float ,int);
Main Begins here
srand(static_cast
<unsigned int>
(time(0)));
//Constants
short const
ENDYR=11
A
A
Define
short plyAgn;
Do
Page 11
AA
True
Page 2
B
Page 2
C
Begin
seeRule();
Page 3
D
Page 1
E
Page 4
F

Get Title


Page 1
B
gtTitle begins
gtTitle();
Define Variables
string ttl;
ofstream title;
title.open("title.dat");
title<<"Hammurabi: A Game Of Strategy";
title.close();
Define
ifstream titleO;
titleO.open("title.dat");
while(titleO>>ttl)
False
True
cout<<ttl<<" ";
titleO.close();
cout<<endl;
End gtTitle
Page 1
C

See Rules



```
Define
char ans;
bool choice;
```

```
cout<<"Press 0 And Enter To See The Game
Rules.\nPress "
"Anything Else To Continue
And Play The Game\n";
```

```
cin>>ans;
```

```
choice=ans-48;
```

```
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";
```

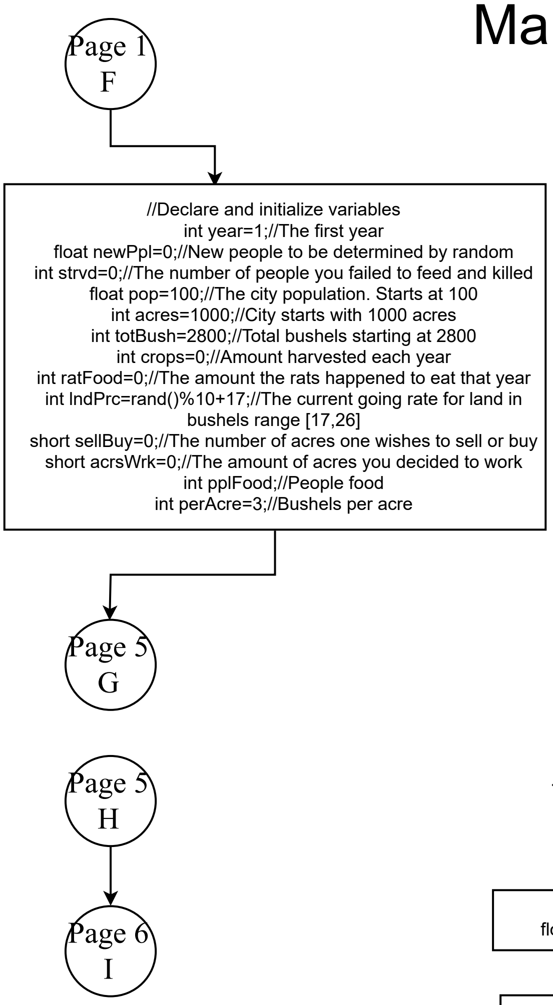
```
cout<<"Press Enter
To Play\n";
```

```
cin.ignore();
cin.get();
```

```
cout<<endl<<endl;
```

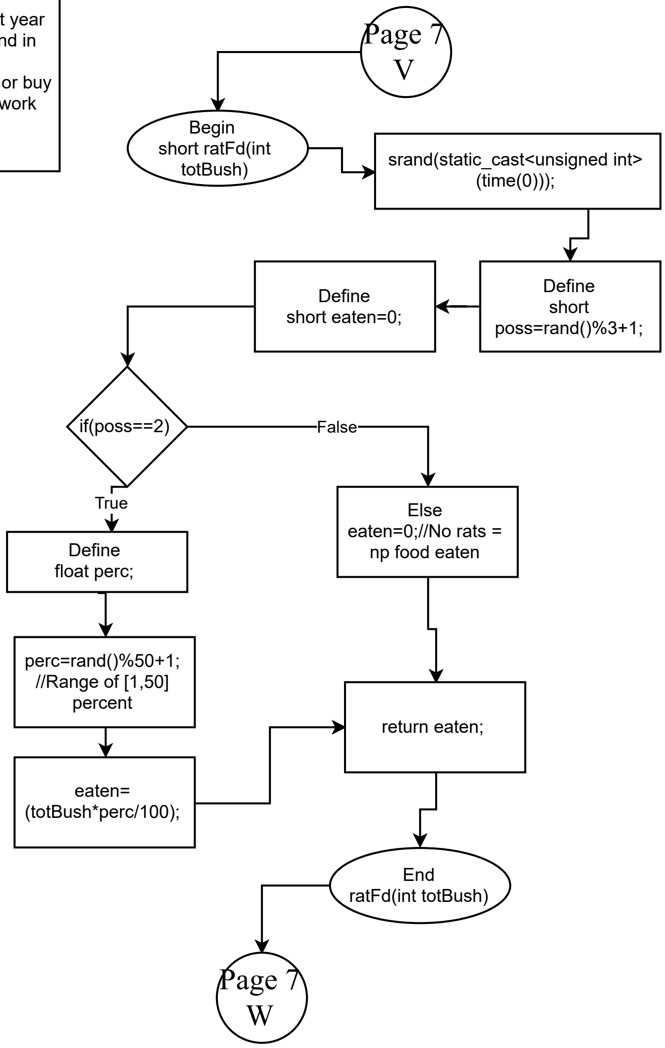
```
End
seeRule();
```

Main Body



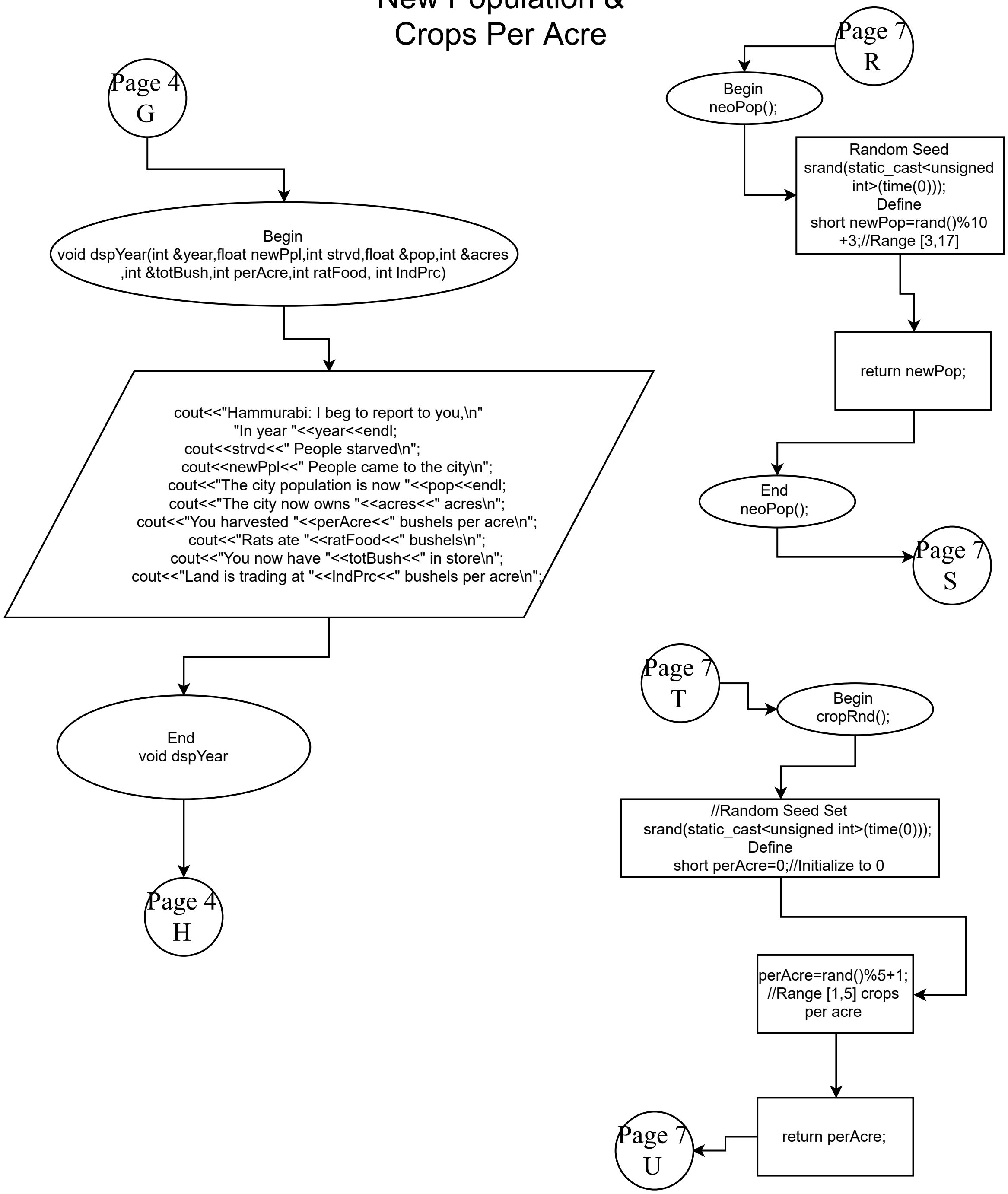
```
//Declare and initialize variables
int year=1;//The first year
float newPpl=0;//New people to be determined by random
int strvd=0;//The number of people you failed to feed and killed
float pop=100;//The city population. Starts at 100
int acres=1000;//City starts with 1000 acres
int totBush=2800;//Total 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
```

Eaten By Rats



```
srand(static_cast<unsigned int>(time(0)));
Define
short
poss=rand()%3+1;
Define
short eaten=0;
if(poss==2)
Define
float perc;
perc=rand()%50+1;
//Range of [1,50]
percent
eaten=
(totBush*perc/100);
Else
eaten=0;//No rats =
np food eaten
return eaten;
```

Display Year & New Population & Crops Per Acre



Begin

```
void dspYear(int &year,float newPpl,int strvd,float &pop,int &acres
,int &totBush,int perAcre,int ratFood, int lndPrc)
```

```
cout<<"Hammurabi: I beg to report to you,\n"
"In year "<<year<<endl;
cout<<strvd<<" People starved\n";
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<<" in store\n";
cout<<"Land is trading at "<<lndPrc<<" bushels per acre\n";
```

End
void dspYear

Begin
neoPop();

Random Seed

```
srand(static_cast<unsigned
int>(time(0)));
```

Define

```
short newPop=rand()%10
+3;//Range [3,17]
```

```
return newPop;
```

End
neoPop();

Begin
cropRnd();

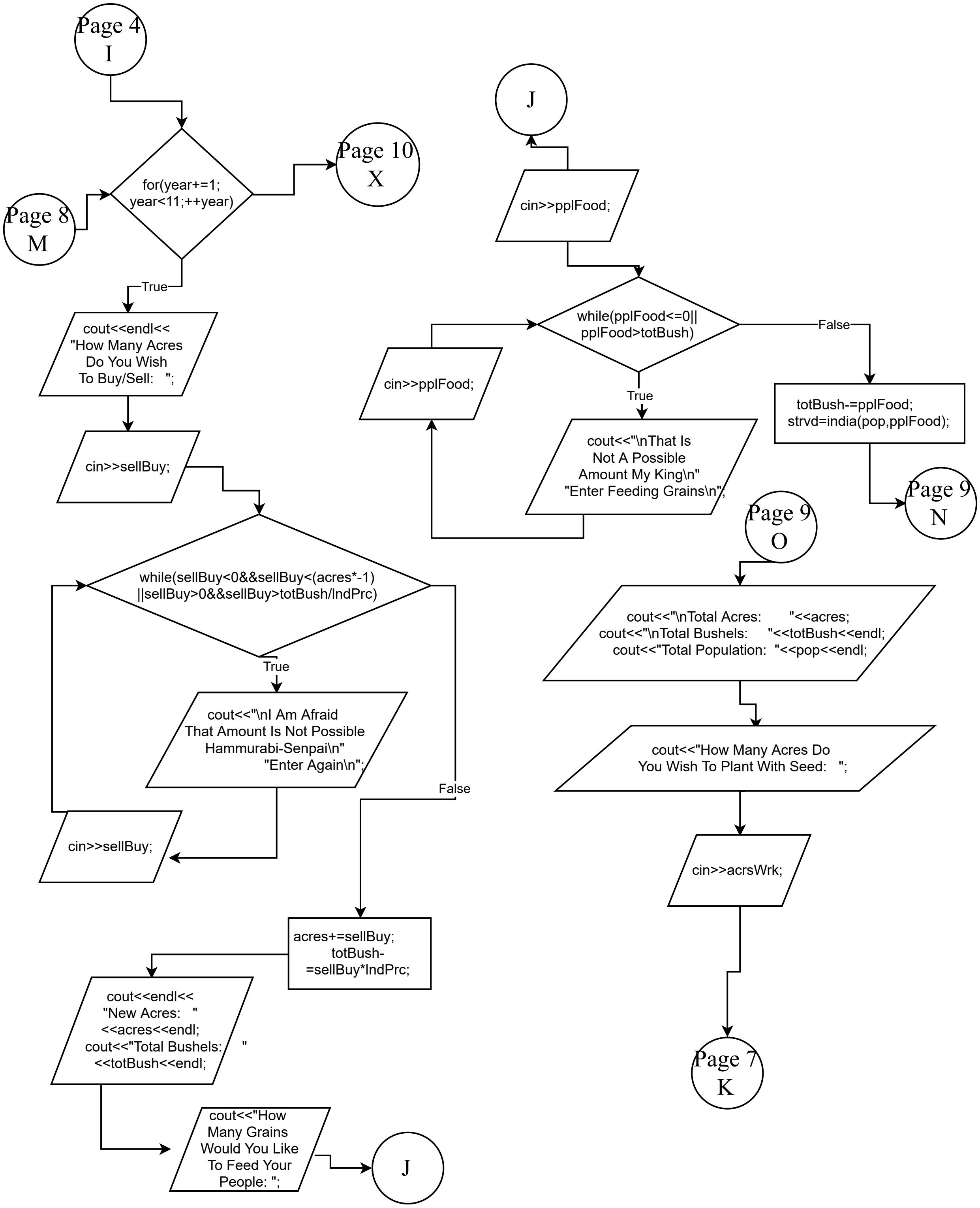
```
//Random Seed Set
srand(static_cast<unsigned int>(time(0)));
```

Define

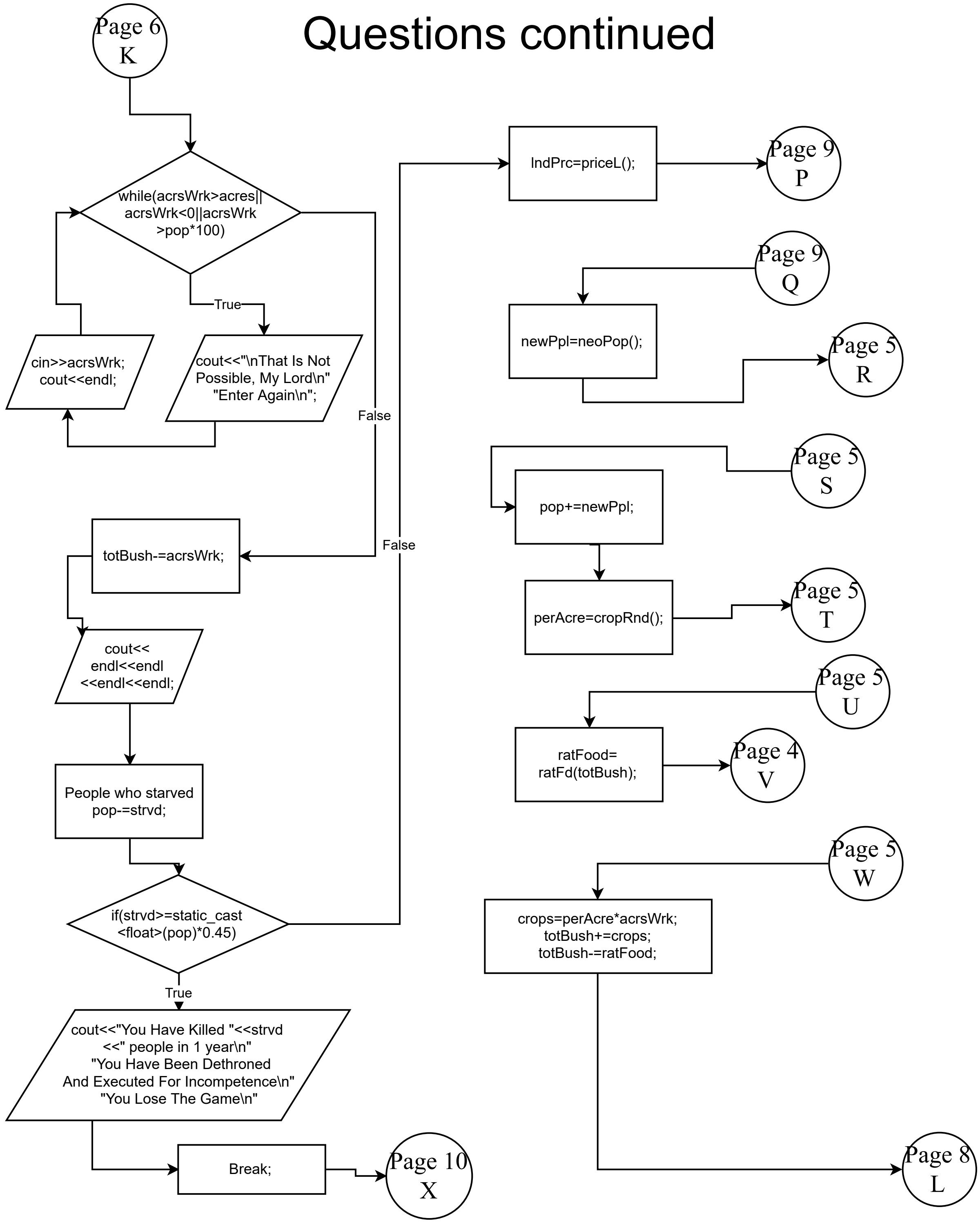
```
short perAcre=0;//Initialize to 0
```

```
perAcre=rand()%5+1;
//Range [1,5] crops
per acre
```

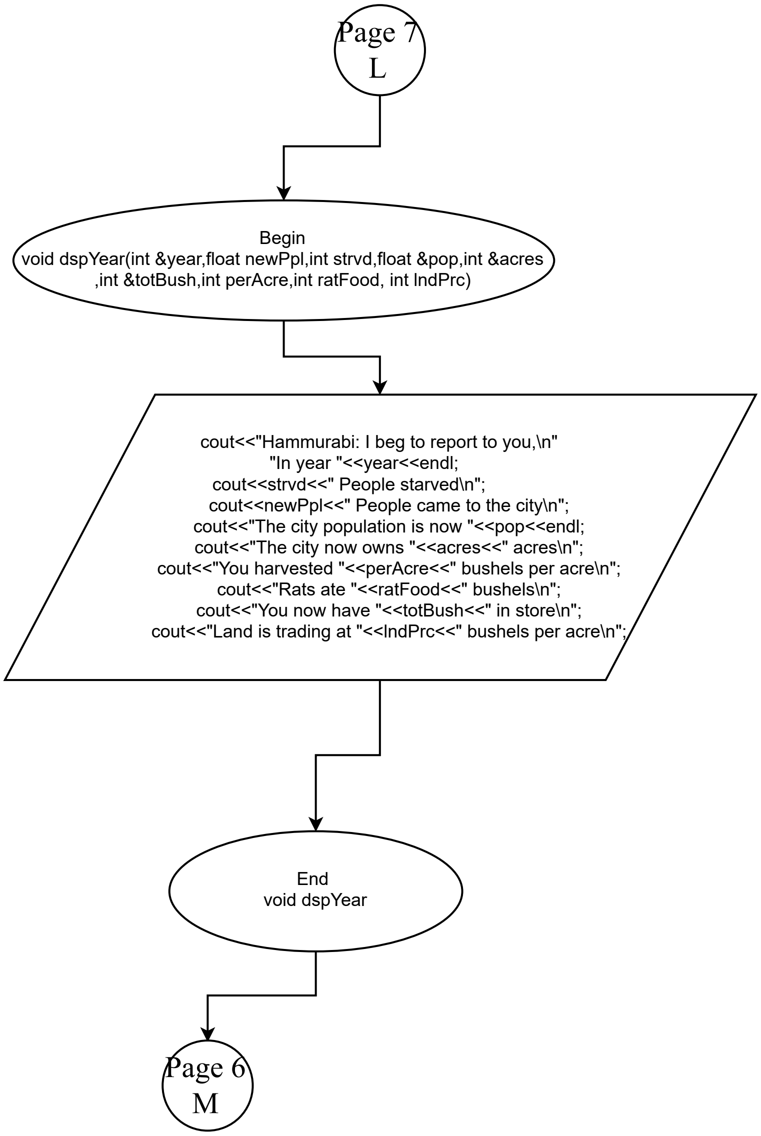
```
return perAcre;
```


Page 4
I
for(year+=1;
year<11;++year)
Page 10
X
Page 8
M
True
cout<<endl<<
"How Many Acres
Do You Wish
To Buy/Sell: ";
cin>>sellBuy;
while(sellBuy<0&&sellBuy<(acres*-1)
||sellBuy>0&&sellBuy>totBush/lndPrc)
True
cout<<"\nI Am Afraid
That Amount Is Not Possible
Hammurabi-Senpai\n"
"Enter Again\n";
False
cin>>sellBuy;
acres+=sellBuy;
totBush-
=sellBuy*lndPrc;
cout<<endl<<
"New Acres: "
<<acres<<endl;
cout<<"Total Bushels: "
<<totBush<<endl;
cout<<"How
Many Grains
Would You Like
To Feed Your
People: ";
J
J
cin>>pplFood;
while(pplFood<=0||
pplFood>totBush)
False
True
cin>>pplFood;
cout<<"\nThat Is
Not A Possible
Amount My King\n"
"Enter Feeding Grains\n";
totBush-=pplFood;
strvd=india(pop,pplFood);
Page 9
N
Page 9
O
cout<<"\nTotal Acres: "<<acres;
cout<<"\nTotal Bushels: "<<totBush<<endl;
cout<<"Total Population: "<<pop<<endl;
cout<<"How Many Acres Do
You Wish To Plant With Seed: ";
cin>>acrsWrk;
Page 7
K

Questions continued



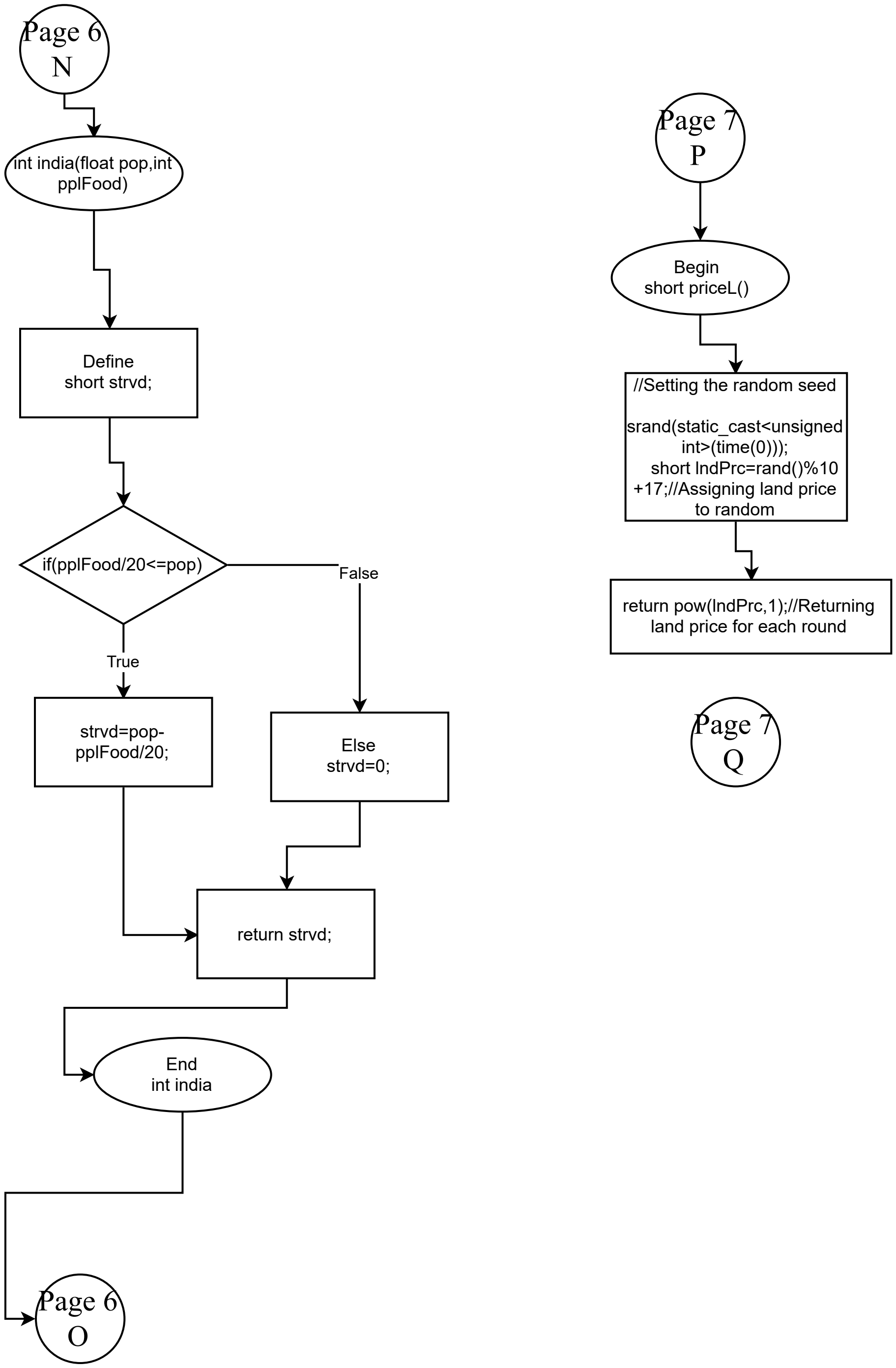
Display Year



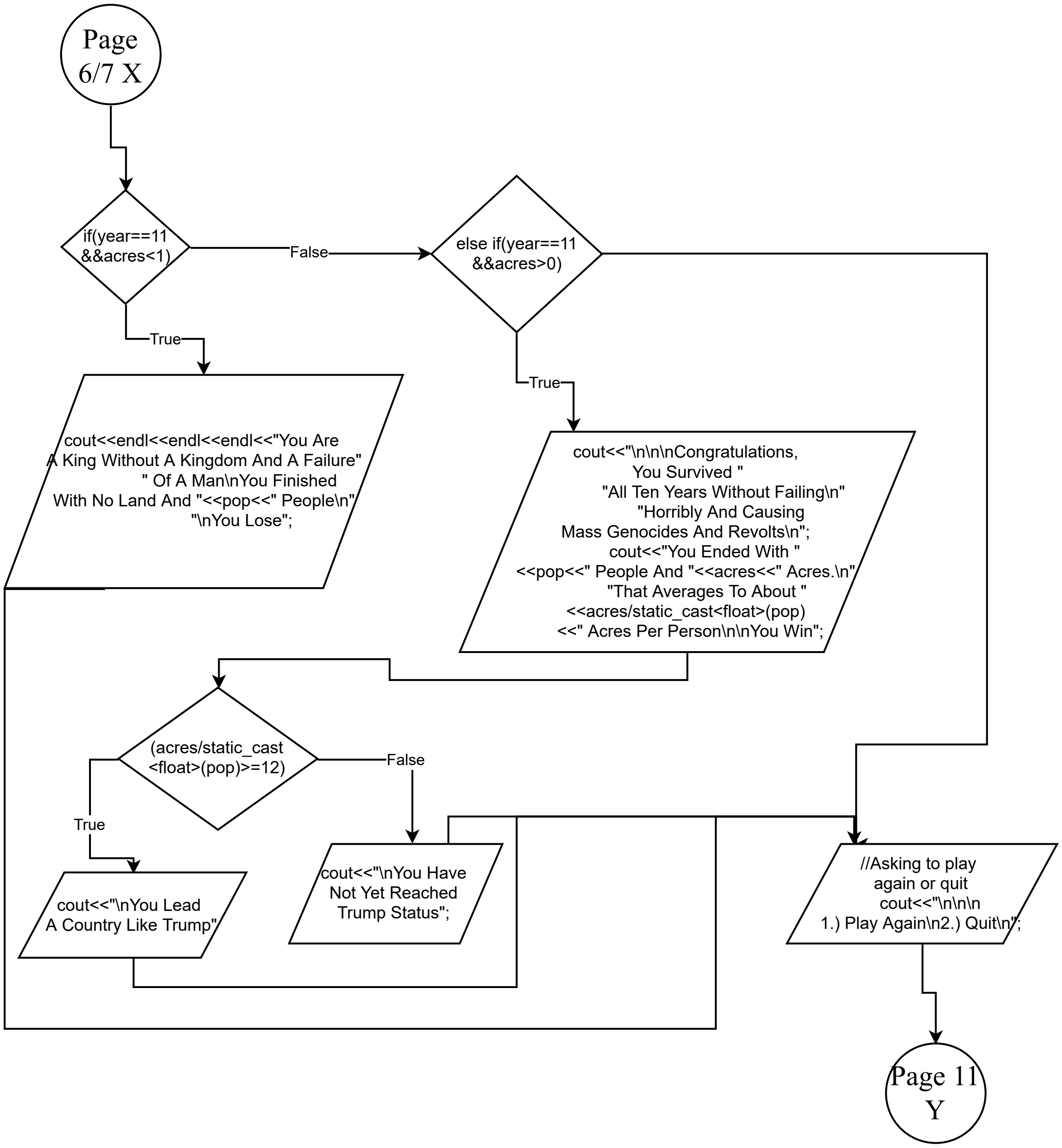
```
void dspYear(int &year,float newPpl,int strvd,float &pop,int &acres
,int &totBush,int perAcre,int ratFood, int lndPrc)
```

```
cout<<"Hammurabi: I beg to report to you,\n"
"In year "<<year<<endl;
cout<<strvd<<" People starved\n";
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<<" in store\n";
cout<<"Land is trading at "<<lndPrc<<" bushels per acre\n";
```

Starving People/ Land Price


Page 6
N
int india(float pop,int pplFood)
Define
short strvd;
if(pplFood/20<=pop)
False
True
strvd=pop-pplFood/20;
Else
strvd=0;
return strvd;
End
int india
Page 6
O
Page 7
P
Begin
short priceL()
//Setting the random seed
srand(static_cast<unsigned int>(time(0)));
short lndPrc=rand()%10 +17;//Assigning land price to random
return pow(lndPrc,1);//Returning land price for each round
Page 7
Q

End Game



```
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";
```

```
else if(year==11
&&acres>0)
```

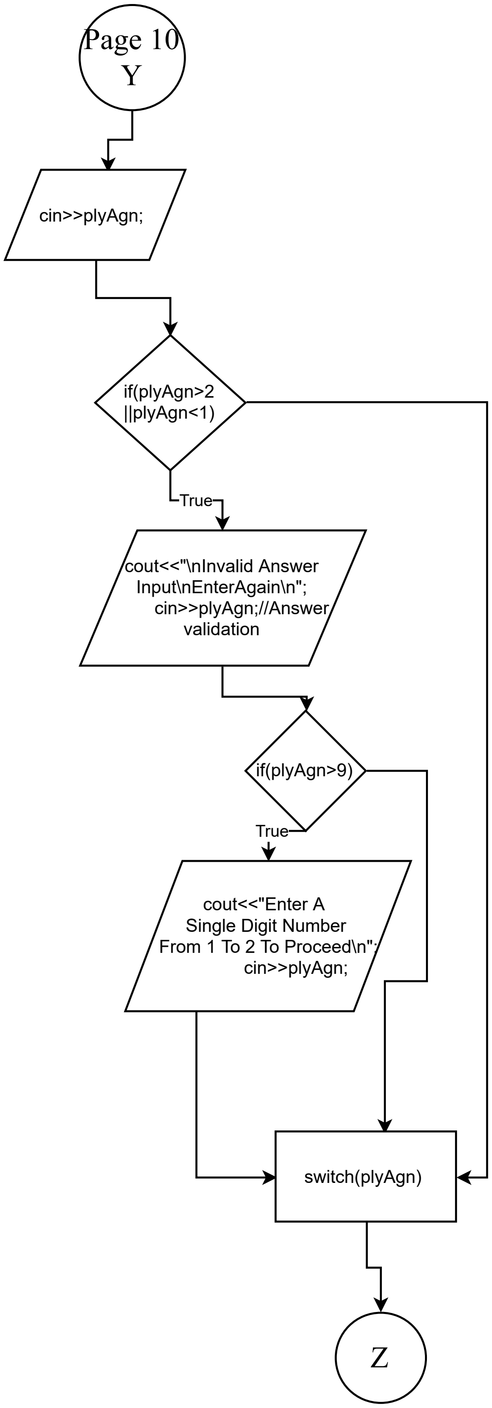
```
cout<<"\n\n\nCongratulations,
You Survived "
"All Ten Years Without Failing\n"
"Horribly And Causing
Mass Genocides And Revolts\n";
cout<<"You Ended With "
<<pop<<" People And "<<acres<<" Acres.\n"
"That Averages To About "
<<acres/static_cast<float>(pop)
<<" Acres Per Person\n\nYou Win";
```

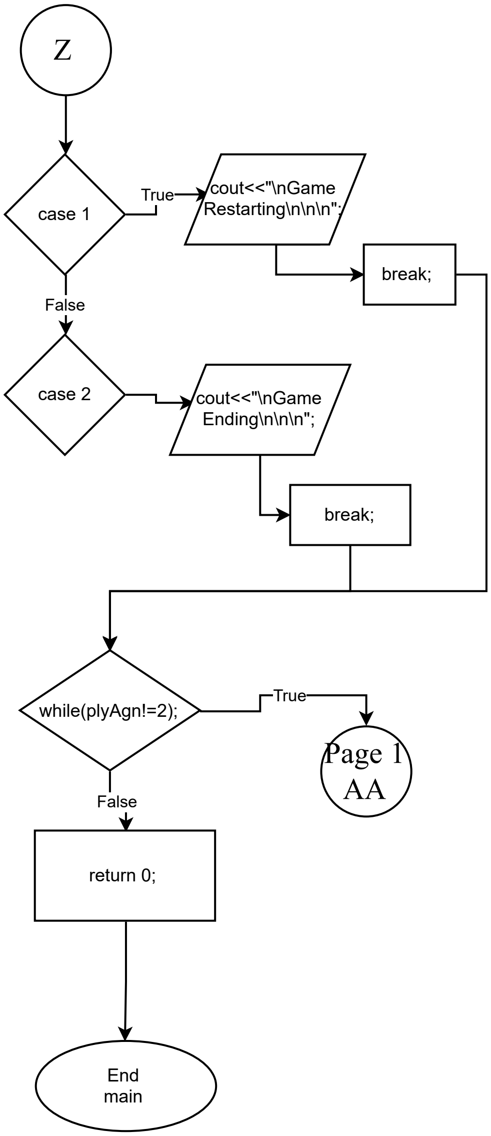
```
(acres/static_cast
<float>(pop)>=12)
```

```
cout<<"\nYou Lead
A Country Like Trump"
```

```
cout<<"\nYou Have
Not Yet Reached
Trump Status";
```

```
//Asking to play
again or quit
cout<<"\n\n\n
1.) Play Again\n2.) Quit\n";
```


Page 10
Y
cin>>plyAgn;
if(plyAgn>2
||plyAgn<1)
True
cout<<"\nInvalid Answer
Input\nEnterAgain\n";
cin>>plyAgn;//Answer
validation
if(plyAgn>9)
True
cout<<"Enter A
Single Digit Number
From 1 To 2 To Proceed\n";
cin>>plyAgn;
switch(plyAgn)
Z


Z
case 1
True
cout<<"\nGame
Restarting\n\n\n";
break;
False
case 2
cout<<"\nGame
Ending\n\n\n";
break;
while(plyAgn!=2);
True
Page 1
AA
False
return 0;
End
main