**Project 1**

**<Guessing Password>**

CSC-17A

Name: Bochi Lin

Date: 11/04/17

**Introduction**

Title: Password Guessing Game

Assume three players are locked three bombs in their bodies. The passwords of these bombs are the same. This game is to get as many points as possible from guessing the password. The bomb with the lowest point will explode. If a player guesses the right password, the bomb will explode as well. The only way to survive is to get as many points as you can. However, if the three players have three same scores or their total score is equal to or greater than 17, they all survive. In round 1 and round 2, each player should check the file after he or she inputs guessing since there will be a hint and score for the player, and he or she should keep the hint as a secret.

**Summary**

Project size: 433 lines

Empty lines: 36 lines

The number of variables: 29

Function Prototypes: 7

**Problems during coding**

It took me about one and a half week to finish this project. Although I have experience doing project, I still had troubles while I was using new concepts to code.

First, I don’t want to make a game only for guessing the right password, but mainly to get points while players are guessing. I spent a few days to come up this idea.

Second, since players are guessing one by one in each round, it is impossible to output hints using “cout” object because I want them to keep the hint as a secret. I tried using functions and delete array methods to do it but it didn’t make sense. At last, I used output file, which is from chapter 12 to show hints and points to players.

**System libraries**

|  |
| --- |
| #include <cstdlib> |
| #include <iostream> |
| #include <ctime> |
| #include <fstream> |
| #include <string> |
| #include <cmath> |
| #include <cstring> |
| #include <cctype> |
| #include <iomanip> |

**Global Constants**

|  |  |  |
| --- | --- | --- |
| const int | SIZE=4 | Digits of number |
| const int | SIZE1=3 | Number of players |

**Structures**

struct PlaInfo{ //Player Information

string name; //Player's name

int score1; //First player score

int score2; //Second player score

int score3; //Third player

};

struct Scores{ //Scores information

int sco1; //Total score for 1st player

int sco2; //Total score for 2nd player

int sco3; //Total score for 3rd player

int total; //Total score for all 3 players

};

struct Number{ //Number digits information

int num;

int digit1; //Digit one

int digit2; //Digit two

int digit3; //Digit three

int digit4; //Digit four

Scores score; //Nested structure Scores

};

**Variables**

|  |  |  |
| --- | --- | --- |
| PlaInfo | plaInfo [SIZE1] | Structure array |
| Number | num | structure |
| int | \*Password  \*Password2 | Two identical  pointer |
| fstream | daFile | Date file |
| int | Score=0 | Initialize score 0 |
| int | number | Call getInput  function |
| int | choice2 | Round 2:  choice |
| int | lowest | Answer for the lowest digit |
| char | answer | Multiple choice |
| int | Number | Random number |
| int | sum1=0  sum2=0  sum3=0  total=0 | In round 1:  First digit  Second digit  Third digit  Fourth digit |
| String  int  int  int | Name  score1  score2  score3 |  |
| int  int  int  int  int  int  int  int  int  Scores | sco1  sco2  sco3  total  num  digit1  digit2  digit3  digit4  score |  |

**Function Prototypes**

|  |
| --- |
| void Introduce( ); |
| int \*getNum ( ); |
| int getInput(Number); |
| void markSrt(int \* ); |
| void outputFile( int, int \* ); |
| bool rightPass( Number, int \* ); |
| bool linearSrch (Number, int \* ); |

**Concepts**

|  |  |  |  |
| --- | --- | --- | --- |
| **concept** | **type** | **code** | **location** |
| Pointer  variable | Pointer  \* | int \*Password; | 60 |
| Pointer  Array  relationship | [ ] --> \* | \*(a+i)=\*(a+i)^\*(a+j);  \*(a+j)=\*(a+i)^\*(a+j);  \*(a+i)=\*(a+i)^\*(a+j); | 392  393  394 |
| Pointer as function  parameters | FunctionName  ( datatype \*) | void markSrt(int \*); | 51 |
| Dynamic  array | iptr =  new int[size]; | int \*ranNum=  new int [SIZE]; | 369 |
| Returning pointers | Return pointer | return ranNum; | 379 |
| Abstract data types | ADT | Number num; | 59 |
| Combining data into structure | struct *tag* {  *variable declaration;*  // ... more declarations  // may follow... }; | truct PlaInfo{  string name;  int score1;  int score2;  int score3;  }; | 24  25  26  27  28  29 |
| Accessing  Structure  members | Name.variable | cin>>plaInfo[i].name; | 72 |
| Array of structures | Ex.  BookInfo bookList[20] | PlaInfo plaInfo[SIZE1]; | 58 |
| Nested structures | struct a{…}  struct b{a…} | Struct Scores {…};  Struct Number {  Scores score  } | 44 |
| Structure as function argument | Ex.  void showRect  (Rectangle r) | int getInput(Number); | 50 |
| File operation | fstream | Fstream daFile | 62 |
|  | ios::out | daFile.open  ("Players.h",ios::out); | 123 |

**Flowchart**

<https://go.gliffy.com/go/publish/12344286>

**Pseudo-code:**

System libraries

//Global Constants

//Structures

//Function Prototypes

Main

//Declare Variables

Introduce the game

//Input information

Input player's name

Get two identical password arrays

//Round 1 begins!

//Input information

Input the number

//Calculation

Get the 4th digit

Get the 3rd digit

Get the 2nd digit

Get the 1st digit

//If statement begins

If player guess the password

Program run finishes

Game over

Return 0

End if

//If statement

if match digits

for loop

if digit matches

output file

End if

End for loop

End if

//Display Output

output grades in round 1

//Round 2 begins

//Input information

input choice2

//Switch statement

case 1: Enter number

If matches

Score add 5

End if

Else

Score add 1

End if

Case 2: Multiple choice

If correct

Score add 2

End if

Else

Score add 0

End if

End switch

//Display output

Output file

Output grades in round 2

//Round 3 begins

//Display output

Output hints

//Input information

Input the password

//If statement

If player guess the password

Program run finishes

Game over

Return 0

End if

//Display output

Output the grades in round 3

//Calculation

Get each score

Get the total score

//Display Output

Output the final score

Output who dies, who survives or all survive

Return 0

End run

**Code:**

/\*

\* File: main.cpp

\* Author: Bochi Lin

\* Created on October 29, 2017, 11:49 AM

\* Purpose: CSC-17A Project 1

\*/

#include <iostream> //I/O

#include <string> //String

#include <cstdlib> //Random

#include <fstream> //I/O file

#include <cmath> //Pow

#include <ctime> //Time

#include <cstring> //Cstring

#include <cctype> //Check upper case

#include <iomanip> //Setw

using namespace std;

//Global Constants

const int SIZE=4; //Digits of number

const int SIZE1=3; //Number of players

//Structures

struct PlaInfo{ //Player Information

string name; //Player's name

int score1; //First player score

int score2; //Second player score

int score3; //Third player

};

struct Scores{ //Scores information

int sco1; //Total score for 1st player

int sco2; //Total score for 2nd player

int sco3; //Total score for 3rd player

int total; //Total score for all 3 players

};

struct Number{ //Number digits information

int num;

int digit1; //Digit one

int digit2; //Digit two

int digit3; //Digit three

int digit4; //Digit four

Scores score; //Nested structure Scores

};

//Function Prototypes

void Introduce(); //Introduce the game

int \*getNum(); //Get the random password

int getInput(Number); //Get the input here

void markSrt(int \*); //Using Mark sort to sort the array

bool linearSrch(int \*,Number); //Linear search

void outputFile(int,int \*); //Output file

bool rightPass(Number,int \*); //Check right password

int main(int argc, char\*\* argv) {

//Declare variables

PlaInfo plaInfo[SIZE1]; //Structure array

Number num; //Structure

int \*Password; //Password pointer

int \*Password2; //Same as Password pointer

fstream daFile; //Data file

int score=0; //Initialize score

//Introduce the game

Introduce();

//Input information

for (int i=0;i<SIZE1;i++){

cout << "Please enter the name of ";

cout << "player " <<i+1<< ": ";

cin>>plaInfo[i].name; //Input player's name

}

//Get two identical password arrays

Password=getNum();

Password2=getNum();

//Rounds begins here!

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* ROUND 1 \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

//Input information

for (int j=0;j<SIZE1;j++) {

cout<<"Player "<<plaInfo[j].name<<", please enter your guess: "<<endl;

int number=getInput(num);

num.digit4=number%10; //Get the 4th digit

num.digit3=(number/10)%10; //Get the 3rd digit

num.digit2=(number/100)%10; //Get the 2nd digit

num.digit1=number/1000; //Get the 1st digit

if(rightPass(num,Password)){ //If player guess the password

cout<<"RESULT: "<<endl;

cout<<"You guess the right password."<<endl;

cout<<"The bomb explodes."<<endl;

cout<<"Game over."; //If the player guess the right password, run finish

return 0;

}

//Round 1 begins here

if (linearSrch(Password,num)){ //Call a bool function here

for (int i = 0; i < SIZE1; i++) {

if (num.digit1 == Password[i]) {

daFile.open("Players.h",ios::out); //Open file

daFile<<"You get 1 point in this round."<<endl;

daFile << num.digit1 << " is right, and it is at digit " << i + 1 << endl;

daFile.close();//close file

}

if (num.digit2 == Password[i]) {

daFile.open("Players.h",ios::out);

daFile<<"You get 1 point in this round."<<endl;

daFile << num.digit2 << " is right, and it is at digit " << i + 1 << endl;

daFile.close();

}

if (num.digit3 == Password[i]) {

daFile.open("Players.h",ios::out);

daFile<<"You get 1 point in this round."<<endl;

daFile << num.digit3 << " is right, and it is at digit " << i + 1 << endl;

daFile.close();

}

if (num.digit4 == Password[i]) {

daFile.open("Players.h",ios::out);

daFile<<"You get 1 point in this round."<<endl;

daFile << num.digit4 << " is right, and it is at digit " << i + 1 << endl;

daFile.close();

}

}

score+=1; //If it is right, add one point

} //Else, the score is still 0

cout<<"Check file 'Players' to see your score!\n"<<endl; //Output score to file

if (j==0) {

plaInfo[0].score1=score; //Add score for the first player

if (score==0) outputFile(plaInfo[0].score1, Password); //If the score is 0, call function

}

if (j==1) {

plaInfo[0].score2 =score; //Add score for the second player

if (score==0) outputFile(plaInfo[0].score2, Password); //If the score is 0, call function

}

if (j==2) {

plaInfo[0].score3=score; //Add score for the third player

if (score==0) outputFile(plaInfo[0].score3, Password); //if the score is 0, call function

}

score = 0; //Initialize score to 0 again

}

//Here are the grades for the first round

//Display output

cout<<"--------- GRADES -----------"<<endl;

cout<< "round 1" << endl;

cout<<" "<<plaInfo[0].score1<<endl;

cout<<" "<<plaInfo[0].score2<<endl;

cout<<" "<<plaInfo[0].score3<<endl;

cout<<"----------------------------"<<endl;

cout<<endl;

//Round 2 begins

markSrt(Password); //Call a function for using Mark sort to sort password array

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* ROUND 2 \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout << "Now. The password is in a new order from low to high."<<endl;

if(Password[0]!=0) cout<<"Please guess the lowest number. "<<endl;

else cout<<"Please guess the number in the second digit. "<<endl; //If the lowest is 0, guess the second digit

cout<<"-------------------------------------------------------"<<endl;

for (int j=0; j<SIZE1;j++) {

//Display menu here

int choice2;

cout<<"Player "<<plaInfo[j].name<<endl;

cout<<"Do you want to guess the number on your own or choose an answer?"<<endl;

cout<<"1.On your own(5 points if correct/1 point if wrong); 2.choose answer(2 points if correct)"<<endl;

cout<<"Please enter your choice: ";

cin>>choice2; //players enter their choice

switch(choice2){

case 1:{

int lowest; //Lowest or 2nd digit

cout<<"==>You chose to guess this number on your own."<<endl;

cout<<"Answer: ";

cin>>lowest;//Enter the number

cout<<"Check file 'Players' to see your score!\n"<<endl; //Output score to file

if(Password[0]!=0){//if the first digit is not 0

if(lowest==Password[0]){

score+=5; //If the number is right, add 5 points

}

else score+=1;//if the number is not right, add 1 point

}

else{//If the first digit is 0

if(lowest==Password[1]){//The number player entered is compare to the second digit

score+=5;//If the number is right, add 5 points

}

else score+=1;//if the number is not right, add 1 point

}

break;

}

case 2:{

char answer;

int Number;

cout<<"==>You choose to guess this number with an multiple choice."<<endl;

Number=rand()%9; //Get a random number here

cout<<"Is "<<Number<<" the number?"<<endl;

cout<<"A.Yes; B.No"<<endl;

cout<<"Answer: ";

cin>>answer; //Player enter his/her choice

cout<<"Check file 'Players' to see your score!\n"<<endl; //Output score to file

if(Password[0]!=0){

if(Number==Password[0]&&answer=='A') score+=2; //If the choice is correct

else if(Number!=Password[0]&&answer=='B') score+=2; //then a 2 points

else score=0;//If the choice is not correct,score is 0

}

if(Password[0]==0){//The same idea as below, but is compare to the second digit

if(Number==Password[1]&&answer=='A') score+=2;

else if(Number!=Password[1]&&answer=='B') score+=2;

else score=0;

}

break;

}

}

daFile.open("Players.h",ios::out);//Open file

daFile<<"You get "<<score<<" points in this round."<<endl; //Output file

daFile.close();//close the file

//Add scores to each arrays of each player

if (j==0) plaInfo[1].score1=score;

if (j==1) plaInfo[1].score2=score;

if (j==2) plaInfo[1].score3=score;

score=0; //Initialize score to 0 again

cout<<endl;

}

//Display the grades here

cout<<endl;

cout<<"---------- GRADES ----------"<<endl;

cout << "round 1 round 2" << endl;

cout<<" "<<plaInfo[0].score1<<setw(11)<<plaInfo[1].score1<<endl;

cout<<" "<<plaInfo[0].score2<<setw(11)<<plaInfo[1].score2<<endl;

cout<<" "<<plaInfo[0].score3<<setw(11)<<plaInfo[1].score3<<endl;

cout<<"-----------------------------"<<endl;

cout<<endl;

//Round 3 begins here

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* ROUND 3 \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Now. The password is back into the original order."<<endl; //Use array Password2

cout<<"Hint: The third digit of the password is "<<Password2[2]<<endl;//Show hints to players

for(int i=0;i<SIZE1;i++){

cout<<"Player "<<plaInfo[i].name<<", please enter your guess:"<<endl;

int number=getInput(num);

num.digit4=number%10; //Get the 4th digit

num.digit3=(number/10)%10; //Get the 3rd digit

num.digit2=(number/100)%10; //Get the 2nd digit

num.digit1=number/1000; //Get the 1st digit

cout<<endl;

if(rightPass(num,Password2)){ //If the player guess the right password, run finish

cout<<"RESULT: "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"You guess the right password.\*"<<endl;

cout<<"The bomb explodes. All die! \*"<<endl;

cout<<"Game over. \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else{

if(pow(num.digit2-Password2[1],2)<=9){//If the range is equal to or less than 3

score+=1; //Score add 1

if(num.digit2==Password2[1]) //If the second digit is correct

score+=3; //Score add 3

}

if(pow(num.digit4-Password2[3],2)<=9){//If the range is equal to or less than 3

score+=2; //Score add 2

}

}

//Add scores to each array

if (i==0) plaInfo[2].score1 = score;

if (i==1) plaInfo[2].score2 = score;

if (i==2) plaInfo[2].score3 = score;

score=0;//Initialize score to 0

}

//Display all three grades

cout<<endl;

cout<<"------------ GRADES ------------"<<endl;

cout<<" round 1 round 2 round3"<<endl;

cout<<" "<<plaInfo[0].score1<<setw(11)<<plaInfo[1].score1<<setw(11)<<plaInfo[2].score1<<endl;

cout<<" "<<plaInfo[0].score2<<setw(11)<<plaInfo[1].score2<<setw(11)<<plaInfo[2].score2<<endl;

cout<<" "<<plaInfo[0].score3<<setw(11)<<plaInfo[1].score3<<setw(11)<<plaInfo[2].score3<<endl;

cout<<"---------------------------------"<<endl;

cout<<endl;

//Calculate the final score

for(int x=0;x<SIZE1;x++){

num.score.sco1+=plaInfo[x].score1;

num.score.sco2+=plaInfo[x].score2;

num.score.sco3+=plaInfo[x].score3;

}

num.score.total=num.score.sco1+num.score.sco2+num.score.sco3;

int sum1=num.score.sco1;

int sum2=num.score.sco2;

int sum3=num.score.sco3;

int total=num.score.total;

cout<<"RESULT:"<<endl;

cout<<plaInfo[0].name<<" has "<<sum1<<" points."<<endl;

cout<<plaInfo[1].name<<" has "<<sum2<<" points."<<endl;

cout<<plaInfo[2].name<<" has "<<sum3<<" points."<<endl;

cout<<"The total point is "<<total<<endl;

if((sum1==sum2&&sum2==sum3)||total>=17) {

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\* You all saved, the bomb has been disposed \*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

if(sum3<sum1&&sum3<sum2) cout<<"Player "<<plaInfo[2].name<<" dies with the lowest score."<<endl;

if(sum2<sum1&&sum2<sum3) cout<<"Player "<<plaInfo[1].name<<" dies with the lowest score."<<endl;

if(sum1<sum2&&sum1<sum3) cout<<"Player "<<plaInfo[0].name<<" dies with the lowest score."<<endl;

if(sum1==sum2){

if(sum2>sum3) cout<<"Player "<<plaInfo[2].name<<" dies with the lowest score."<<endl; //3rd lowest

else cout<<"Player "<<plaInfo[2].name<<" survives with the highest score."<<endl; //3rd highest

}

if(sum1==sum3){

if(sum3>sum2) cout<<"Player "<<plaInfo[1].name<<" dies with the lowest score."<<endl; //2nd lowest

else cout<<"Player "<<plaInfo[1].name<<" survives with the highest score."<<endl; //2nd highest

}

if(sum3==sum2){

if(sum2>sum1) cout<<"Player "<<plaInfo[0].name<<" dies with the lowest score."<<endl; //1st lowest

else cout<<"Player "<<plaInfo[0].name<<" survives with the highest score."<<endl; //1st highest

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

//Program ends

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function \*

// Prototypes \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Introduce the whole game here

void Introduce(){

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\*\* Welcome to the Passwords Guessing Game \*\*\n";

cout<<"\*\* This game is to find password for the bomb \*\*\n";

cout<<"\*\* Assume you three players have bombs lock your bodies \*\*\n";

cout<<"\*\* The passwords are the same \*\*\n";

cout<<"\*\* You should try your best to get points in three rounds \*\*\n";

cout<<"\*\* If you get the right password, don't think you are lucky \*\*\n";

cout<<"\*\* because the bomb will explode \*\*\n";

cout<<"\*\* What you have to do is getting as many points as you can \*\*\n";

cout<<"\*\* The person with the lowest points will die \*\*\n";

cout<<"\*\* Make sure to check file after you input in round 1 and 2 \*\*\n";

cout<<"\*\*If the total point is >=17 or 3 scores are same you all survived\*\*\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

}

//Get the random password here

int \*getNum(){

//Declare variables

int \*ranNum=new int [SIZE]; //Set a dynamic array

//Set a random seed

srand(static\_cast<int> (time(0)));

//Get a four-digit random password here

for (int i = 0; i < SIZE; i++) {

ranNum[i]=rand()%9;

}

//Return the array

return ranNum;

}

int getInput(Number n){

cin>>n.num;

return n.num;

}

//Mark sort to sort password digits from low to high

void markSrt(int \*a){

for(int i=0;i<SIZE-1;i++){

for(int j=i+1;j<SIZE;j++){

if(\*(a+i)>\*(a+j)){

\*(a+i)=\*(a+i)^\*(a+j);

\*(a+j)=\*(a+i)^\*(a+j);

\*(a+i)=\*(a+i)^\*(a+j);

}

}

}

}

//Linear search to find the right digit

bool linearSrch(int \*a, Number n) {

for (int i=0;i<SIZE;i++) {

if (n.digit1==a[i]) return true;

if (n.digit2==a[i]) return true;

if (n.digit3==a[i]) return true;

if (n.digit4==a[i]) return true;

}

return false;

}

//Output hints to file called "Players.h" for each player

void outputFile(int a, int \*b) {

fstream daFile;

if (a==0){

daFile.open("Players.h");

daFile<<"You get 0 points in this round."<<endl;

daFile<<"Here is a hint for you: "<<endl;

daFile<<"The first digit of the password is "<<\*b<<endl;

daFile.close();

}

}

//Determine whether the player enter the right password or not

bool rightPass(Number n,int \*a){

if(n.digit1==a[0]){

if(n.digit2==a[1]){

if(n.digit3==a[2]){

if(n.digit4==a[3]) return true;

}

}

}

return false;

}