Project 1

<Guessing Password>

<Version 2.0>

Name: Haolan Ye (Benjamin)

Class: CSC-5 40717

Date: 02/11/2015

Contents

1. Inspiration…….....2

2. Introduction……...2

3. Summary…….....3

4. Problems during coding……3

5. Pseudo Code……..…5

6. Screen shot…….…….…….....6

7. System Libraries……..….9

8. Variable List……..……..…….9

9. Function List...……..……...11

10. Concept Covered…..…….…….…….....12

11. Flowchart...……..……..14

12. Code...……..……..22

1. Inspiration:

This game is to guessing a 4 or 6-digit password. The first inspiration of making this game is from Hangman (A game of guessing words). The second inspiration is from a 4-digit coded lock of my luggage. After I travelled to a city, I forgot the password of my lock. Then, I tried lots of times to guess the password. Every time after I guessed, I need to try to unlock it.

2. Introduction:

Different from Hangman, players should guess the numbers instead of words in Guessing Password. There are three different levels (Easy, Medium, and Hard) for players to select. First, according to the level that players choose, computer will randomly generate a 4-digit number (from 0000 to 9999) for level 1 & 2, 6-digit number (from 000000 to 999999) for level 3. Then, for every level, players only can guess one digit of password at one time. Players should input a number they guess and the digit of this number in order. Afterward, the computer will tell you whether the number and digit are wrong or not. Players only have 12 (level 1&2)/18 (level 3) chances to guess the password. When the player guesses the number correctly, the chances left won't count down. At the end of the game, Top 10 records will be displayed.

On Easy level, when players do not know how to play this game, they can type "s" and then press Enter to ask the computer to display the sample input of this game. On Medium and Hard level, players have one time to type "h" if they want to get the

hint for a digit of password. The hint will be changed to binary number. Players cannot guess the same number in a specific digit after this digit has been done.

3. Summary:

Total Line of Code	377
Blank Line	8
Comment Line	33
Variable	26
Function	13

This game contains most concepts that we have learned in the class. When I made this game, I found lots of problems that I haven't thought about. I came up with the structure of this game in the 2nd week of the class. Even though I had structure early, I spent most of time to solve some difficult problems of my game. Different from Version 1, version 2 has 3 different levels for players to select. In addition, version 2 can record how long players finished their game. At the end of the game, Top 10 Records will be outputted.

4. Problems during coding

a) **Cannot generate the password that begins with 0**

At the beginning of coding this game, I used integer variables to store the password, but when I output the password, I found computer cannot generate the integer begins with 0. Therefore, I used char array to store the password. First, randomly generate char '0' to '9', and then store it into array.

The advantages of char array for password:

1) The computer can check whether the guess is correct more easily
2) Every digit of password can store at the specific place in the array

b) Do not know how to check the char whether is the char array

Before taking this class, I learned Java on YouTube by myself. In Java, there is a function that can return the index of the char in the string. But in C++, I do not know how to get the index of the char from string. Afterward, I wrote a function with for loop that returns the index of char.

c) Record the digits that have been done

When I test the game a couple days ago, after I guessed a digit of password correctly, I tried to input the same guess. The game still display I have guessed correct. After I input the same correct digit for 4 times, I won the game. Therefore, I have to record the digits that have been guessed correctly. Otherwise, the game will have a huge bug. I use vector to record them. After player guessed a digit of password correctly, that digit will be stored in the integer vector. Afterward, when players tried to input the same correct answer at one digit, the computer will remind the players.

d) Hint

In order to make the game easy for everyone, I tried to give the hint to players. After I did that, I found the hint could be displayed every time that players type “h”. Also, giving the accurate number of a digit is so easy for everyone. Therefore, I create a new variable to keep track of how many hints used. And then, I write a new function to display the hint in binary number.

5. Pseudo Code

Introduce the game

Prompt players for level selection

According to level that player selects, randomly generate password, stores the copy of the password in string

Create dash (string)

Get the current time

When (chance left>0 and the number of correct guesses<size of password)

Do (

Display the dash and chances left

Prompt user for guess (Easy level, if players need sample input, type “s”)

(Medium and Hard level, if players need hint, type “h”)

) Until players input number and digit correctly

Check whether the number and digit are correct or not

If the number and digit both are correct

Replace the digit of dash

If this digit has been guessed correctly, remind the player

Else record this digit, and then correct guess +1

If number is correct but digit is not

Display the result of this guess

Chance left -1

If number and digit are both wrong

Display the result of this guess

Chance left -1

If player wins the game, save the playing time to file, and sort the file

Output the result of this game and output to a file

Display Top 10 Records

6. Screen Shot

(1) Select the level

```
*****************LEVEL SELECTION*****************
    1.Easy:
      Feature: (1)4 digit
               (2)No Repeat Number
               (3)Including sample input
    2.Normal:
      Feature: (1)4 digit
               (2)Might has repeat number
               (3)Including a hint for a digit
    3.Hard
      Feature: (1)6-digit
               (2)Might has repeat number
               (3)Including a hint for a digit
Please choose a level(Type 1,2,or 3)
1
```

(2) Guess the number (Wrong guess)

```
The password now looks like this: ----
You have 12 chances left
Please input a 1-digit number you guess(0-9)
If you need sample for input, type 's'
1
Please input the digit of this number(1-4)
1
Wrong number and wrong place.

The password now looks like this: ----
You have 11 chances left
Please input a 1-digit number you guess(0-9)
If you need sample for input, type 's'
```

(3) Guess number (correct number wrong place)

```
The password now looks like this: ----
You have 8 chances left
Please input a 1-digit number you guess(0-9)
If you need sample for input, type 's'
5
Please input the digit of this number(1-4)
1
This is the correct number but in wrong place.
```

(4) Guess number (correct)

```
The password now looks like this: 76-5
You have 3 chances left
Please input a 1-digit number you guess(0-9)
If you need sample for input, type 's'
9
Please input the digit of this number(1-4)
3
Your guess is correct.
You win this game after 13 tries
```

(5) Display the Record

```
The answer is 7695
You spent 70 seconds to play this game

*********Top 10 Records*********
  Rank       Time(seconds)
    1             20
    2             28
    3             54
    4             67
    5             70
  ----          ----
  ----          ----
  ----          ----
  ----          ----
  ----          ----
```

(6) Display the sample input

```
*******************************Sample*********************************
Now the password looks like this: ----
If you guess the number is 4 and it is in the second digit from left
You should input 4 first, then input 2
Then, the computer will tell you whether the number and digit are correct
< This is the end of the sample, continue playing the game>

Please input a 1-digit number you guess(0-9)
If you need sample for input, type 's'
```

(7) Display the hint

```
The password now looks like this: ----
You have 12 chances left
Please input a 1-digit number you guess(0-9)
If you need hint for answer, type 'h'
h
Digit 3 in binary is 100
Please input a 1-digit number you guess(0-9)
```

7. System Libraries

- ✓ **<iostream>**
- ✓ **<cstdlib>**
- ✓ **<string>**
- ✓ **<vector>**
- ✓ **<fstream>**
- ✓ **<iomanip>**

8. Variables List

Type	Variable Name	Description	Declare Location(line)
ofstream	output	Output to the file	42,344
const int	TOTCHNS1	Total chance for level 1&2	46
	TOTCHNS2	Total chance for level 3	47
	SIZE1	Size of the password (level 1&2)	48
	SIZE2	Size of the password (level 3)	49
	COL	Column of 2 dimension array	20
string	dash	Password look like	50
	answer	Answer of password	51
	dashed	String that returns to dash	160
	part1	Substring from the first digit to one digit before guess	231
	part2	Substring from one digit after guess to the end of password	232
	str	Store the string that comes from file	246

int	size	Size of password depends on level	52
	level	Level of the game player select	53
	tyTime	Times player tried	54
	strTime	The time when game begins	55
	endTime	The time when game end	56
	usdTime	End time – starting time	57
	gusCorr	Number of correct guess	58
	chnsLft	How many chance left	59
	hin	Hint of a digit of password	60
	digit	The digit of number that player guess	61
	temp	Temp for int	284
	tip	The tip that will be display	285
	rm	Remainder	305
	rs	Result for conversion to binary	306
char	guess	The number player guess	62
	pswd[SIZE2]	The password that store in char array	63
	temp	Temp for random number	262
	result	The result of one guess	111
vector<int>	inputDg	Done digits	64
	ary	For binary conversion	308
	tmRecd	Record of the time	322
ifstream	input	Input the file	247,358
boolean	temp	Temp for boolean	222,237,259,286
int[][]	rc	Records of player	356

9. Function List

Type	Name	Argument	Function	Location
string	toDash	int	Get the dash for display	159
void	introduce	None	Introduce the game	167
void	gtPswd	char[],int,const int,const int	According to level, generate password	258
void	ask	char&,int&, int, int vector<int>,char[],int&	Prompt player for number and digit	181
char	check	char, const char[],int	check whether number and digit are correct	210
bool	indexOf	char, const char[],int	return whether the char is in the char array	221
void	replace	string&,char,int	replace of the correct digit	230
bool	inside	const vector<int>,int	return whether this digit is finished	236
void	sample	None	display the sample of guessing	245
void	hint	vector<int>,char[],int	display the hint for players	283
int	bin	int	Change a number to binary	304
void	record	int	save the top 10 player record	321
void	dspRecd	None	Display the top 10 record	352

10. Concept covered

Concept	Type	Code	Location(line)
Input / Output	cout	cout<<endl	70
	cin	cin>>guess	188
Variables	int, char,string, ifstream,ofsream	int level	53
Comment		//system Libraries	8
Type casting	static_cast<type>	static_cast<unsigned short>(time(0))	40
Making decisions	if	if(level==1)	184
	if-else	if(ipDg.size()!=0) else	284

	if-else-if	if(guess==pswd[digit-1])
		else if(indexOf(guess,pswd,size))
		else
			212

	switch	switch(result)	112
Loop	for	for(int i=0;i<size;i++)	161
	while	while(chnsLft>0&&gusCorr<size)	105
	do-while	do {} while (guess<48||guess>57)	182
File I/O	ifstream	ifstream input; input.open("Sample.dat");	247
	ofstream	ofstream output; output.open("Record.dat");	344

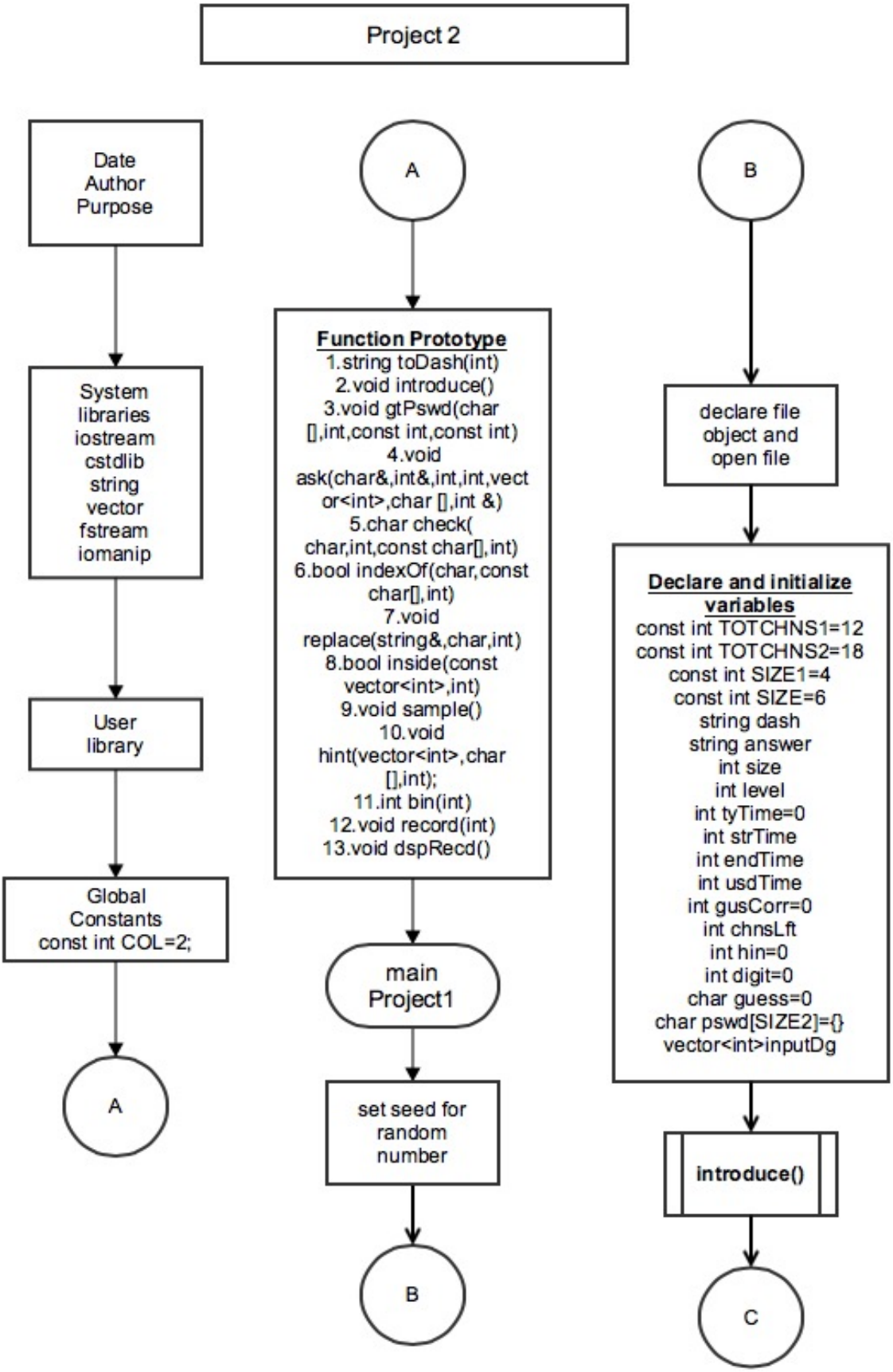
Function	void, int, string, char, bool	string toDash(int)	23
Array	char[]	char pswd[SIZE2]={}	63
	int [][]	int rc[ROW][COL]={}	356
vector	vector<int>	vector<int> inputDg	64

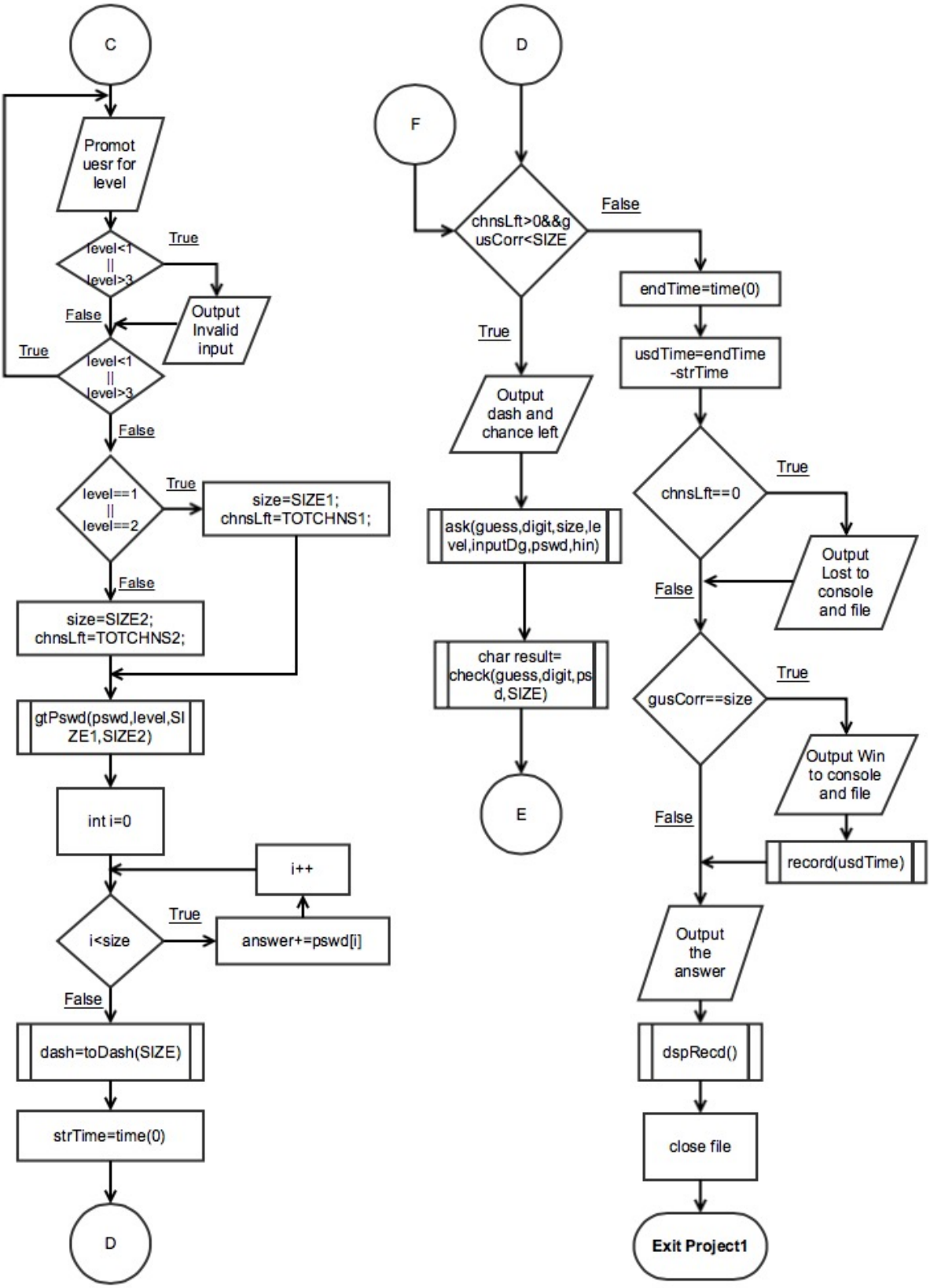
Sorting	Sorting	for(int i=0;i<tmRecd.size()-1;i++) {
		for(int j=i+1;j<tmRecd.size();j++) {
		if(tmRecd[i]>tmRecd[j]) {
		int temp=tmRecd[i];
		tmRecd[i]=tmRecd[j];
		tmRecd[j]=temp;
		}
		}
		}
			334

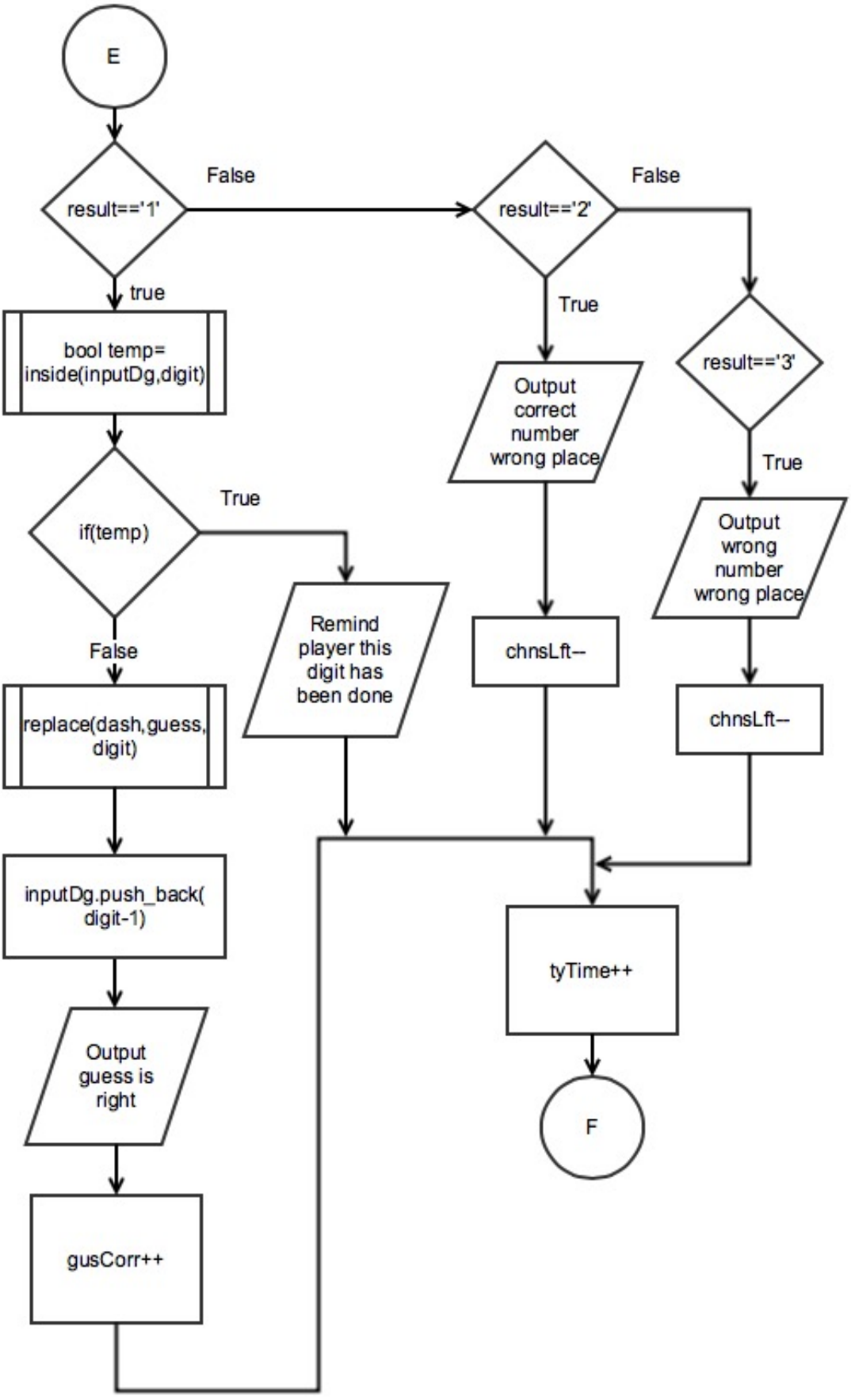
Search	search	bool temp=false;
		for(int i=0;i<inputDg.size();i++) {
		if(inputDg[i]==(digit-1))
		temp=true;
		}
		return temp;
			237

Formatt Output	setw()	cout<<setw(5)<<rc[i][j]<<" ";	371
Random number	char	srand(static_cast<unsigned short>(time(0))); temp=rand()%10+'0';	40,290
Pass array to function	function	void gtPswd(char [],int,const int,const int)	25

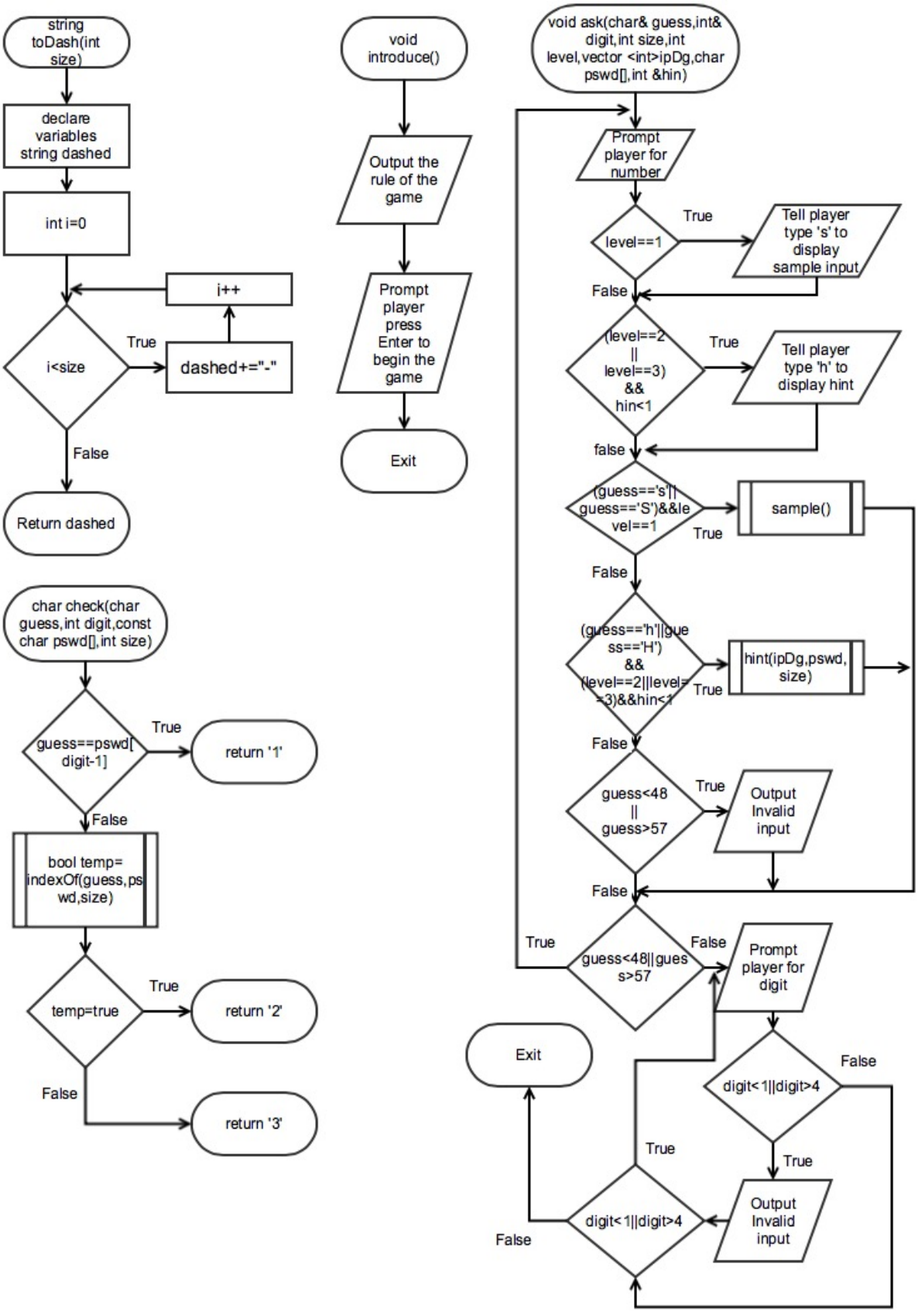
11. Flowchart (1)main flowchart(3 pages)

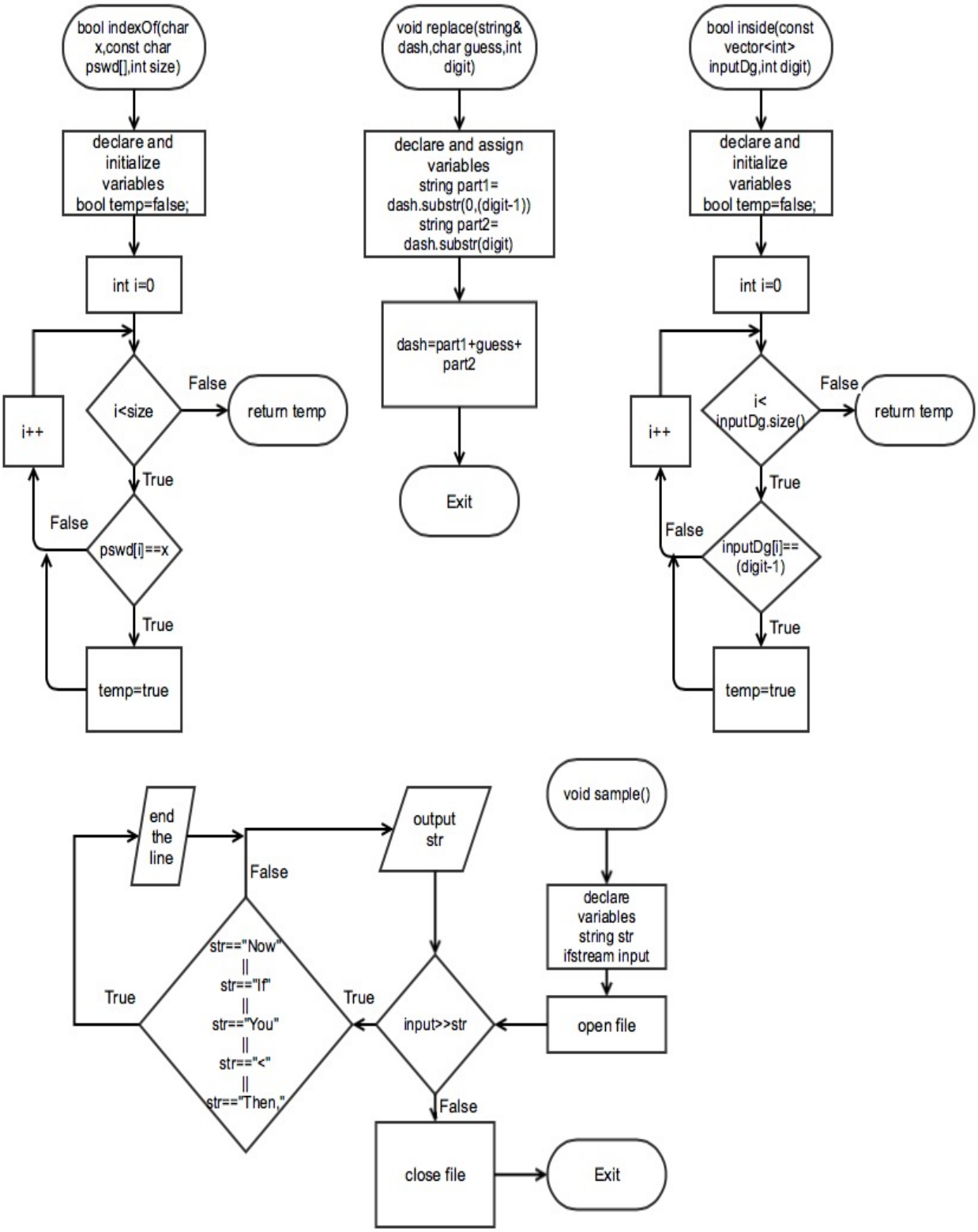


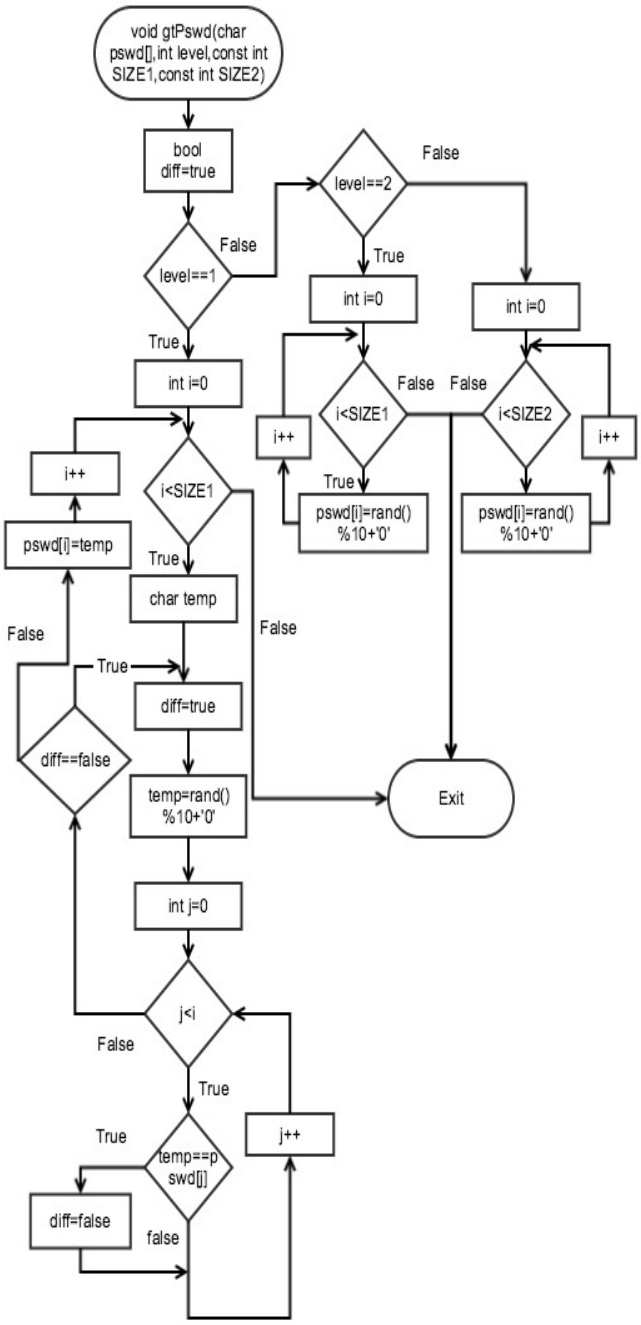

C
Promot uesr for level
level<1 || level>3
True
Output Invalid input
False
level<1 || level>3
True
False
level==1 || level==2
True
size=SIZE1; chnsLft=TOTCHNS1;
False
size=SIZE2; chnsLft=TOTCHNS2;
gtPswd(pswd,level,SIZE1,SIZE2)
int i=0
i++
i<size
True
answer+=pswd[i]
False
dash=toDash(SIZE)
strTime=time(0)
D
D
F
chnsLft>0&&gusCorr<SIZE
False
True
Output dash and chance left
ask(guess,digit,size,level,inputDg,pswd,hin)
char result= check(guess,digit,psd,SIZE)
E
endTime=time(0)
usdTime=endTime -strTime
chnsLft==0
True
Output Lost to console and file
False
gusCorr==size
True
Output Win to console and file
False
record(usdTime)
Output the answer
dspRecd()
close file
Exit Project1

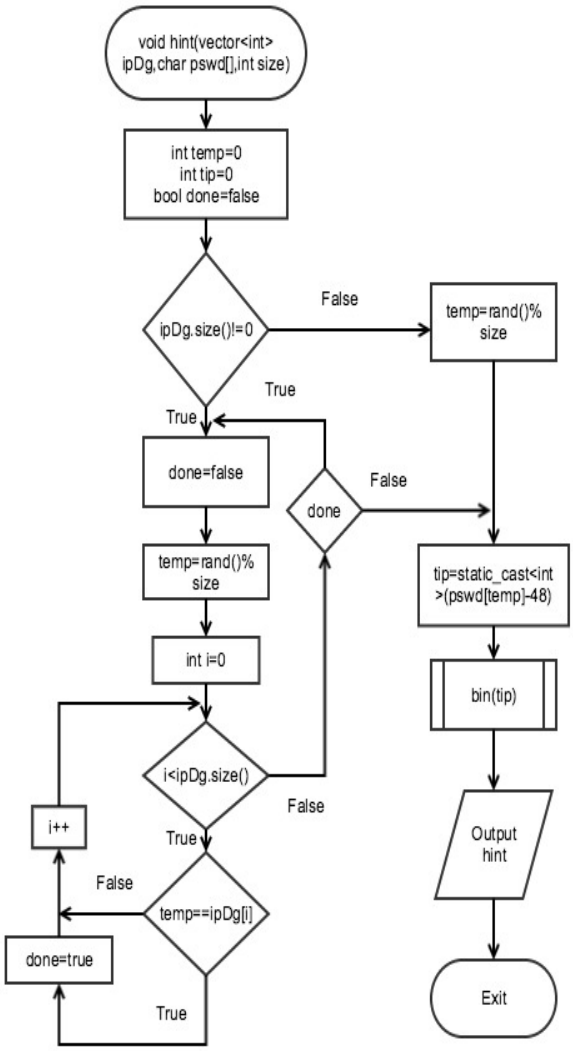

E
result=='1'
False
result=='2'
False
true
True
bool temp=
inside(inputDg,digit)
Output
correct
number
wrong place
result=='3'
True
if(temp)
True
Output
wrong
number
wrong place
Remind
player this
digit has
been done
chnsLft--
False
replace(dash,guess,
digit)
chnsLft--
inputDg.push_back(
digit-1)
tyTime++
Output
guess is
right
F
gusCorr++

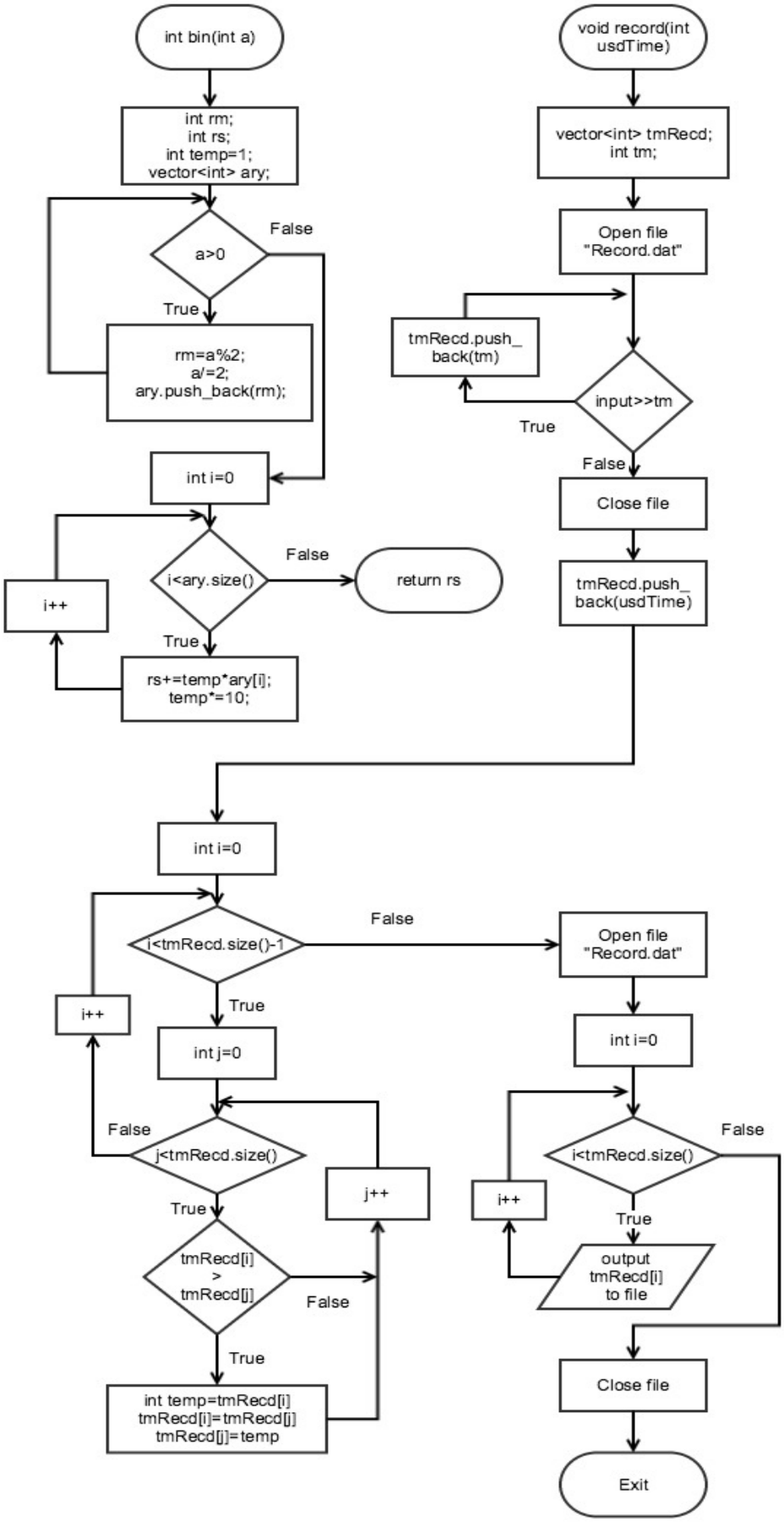
(2) Function flowchart(5 pages)

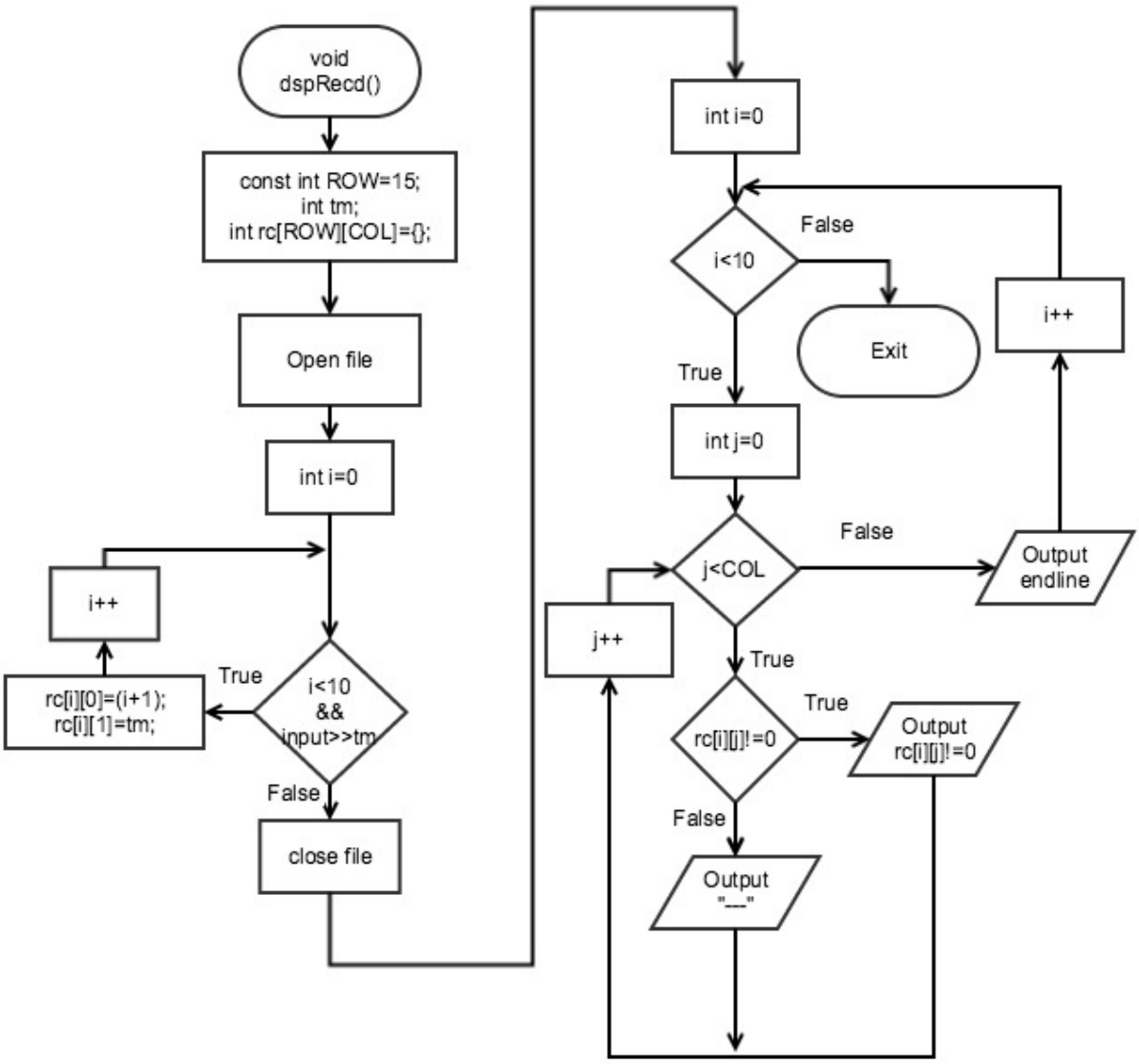



bool indexOf(char x,const char pswd[],int size)
declare and initialize variables bool temp=false;
int i=0
i<size
False
return temp
True
pswd[i]==x
False
True
temp=true
i++
void replace(string& dash,char guess,int digit)
declare and assign variables string part1= dash.substr(0,(digit-1)) string part2= dash.substr(digit)
dash=part1+guess+ part2
Exit
bool inside(const vector<int> inputDg,int digit)
declare and initialize variables bool temp=false;
int i=0
i< inputDg.size()
False
return temp
True
inputDg[i]== (digit-1)
False
True
temp=true
i++
void sample()
declare variables string str ifstream input
open file
input>>str
False
True
str=="Now" || str=="If" || str=="You" || str=="<" || str=="Then,"
True
False
output str
end the line
close file
Exit


void gtPswd(char pswd[],int level,const int SIZE1,const int SIZE2)
bool diff=true
level==1
False
level==2
True
False
int i=0
int i=0
int i=0
True
i<SIZE1
False
False
i<SIZE1
i<SIZE2
i++
i++
i++
True
True
pswd[i]=temp
pswd[i]=rand() %10+'0'
pswd[i]=rand() %10+'0'
True
char temp
False
False
True
diff=true
diff==false
temp=rand() %10+'0'
Exit
int j=0
j<i
False
True
True
temp==p swd[j]
j++
diff=false
false


void hint(vector<int> ipDg,char pswd[],int size)
int temp=0
int tip=0
bool done=false
ipDg.size()!=0
False
temp=rand()% size
True
True
done=false
done
False
temp=rand()% size
tip=static_cast<int >(pswd[temp]-48)
int i=0
bin(tip)
i<ipDg.size()
False
Output hint
i++
True
False
temp==ipDg[i]
done=true
Exit
True


int bin(int a)
int rm;
int rs;
int temp=1;
vector<int> ary;
a>0
False
True
rm=a%2;
a/=2;
ary.push_back(rm);
int i=0
i<ary.size()
False
return rs
True
rs+=temp*ary[i];
temp*=10;
i++
void record(int usdTime)
vector<int> tmRecd;
int tm;
Open file "Record.dat"
input>>tm
True
tmRecd.push_back(tm)
False
Close file
tmRecd.push_back(usdTime)
int i=0
i<tmRecd.size()-1
False
True
int j=0
j<tmRecd.size()
False
True
tmRecd[i] > tmRecd[j]
False
True
int temp=tmRecd[i]
tmRecd[i]=tmRecd[j]
tmRecd[j]=temp
j++
i++
Open file "Record.dat"
int i=0
i<tmRecd.size()
False
True
output tmRecd[i] to file
i++
Close file
Exit


void
dspRecd()
const int ROW=15;
int tm;
int rc[ROW][COL]={};
Open file
int i=0
i<10
&&
input>>tm
True
rc[i][0]=(i+1);
rc[i][1]=tm;
i++
False
close file
int i=0
i<10
False
Exit
True
int j=0
j<COL
False
Output
endline
i++
j++
True
rc[i][j]!=0
True
Output
rc[i][j]!=0
False
Output
"—"

12. Code

```
/*
 * File:   main.cpp
 * Author: Haolan Ye(Benjamin)
 * Created on January 27, 2015, 12:32 PM
 * Purpose: Project2(Name of the game:Bomb Password)
 */

//system Libraries
#include <iostream>
#include <cstdlib>  //for random number
#include <string>
#include <vector>
#include <fstream>  //file I/O
#include <iomanip>
using namespace std;

//User Libraries

//Global Constants
const int COL=2;//For 2-dimension array

//Function Prototypes
string toDash(int);//change the password to dash
void introduce();//introduce the game
void gtPswd(char [],int,const int,const int);
```

```
void ask(char&,int&,int,int,vector<int>,char [],int &);//ask user for guessing
char check(char,int,const char[],int);//check whether number and digit are correct
bool indexOf(char,const char[],int);//return whether the char is in the char array
void replace(string&,char,int);//replace of the correct digit
bool inside(const vector<int>,int);//return whether this digit is finished
void sample();//display the sample of guessing
void hint(vector<int>,char [],int);//display the hint for players
int bin(int);//change an integer to binary
void record(int);//save the top 10 player record
void dspRecd();//Display the top 10 record

//Execution begins here
int main(int argc, char** argv) {
    //set seed for random number
    srand(static_cast<unsigned short>(time(0)));
    //declare a file object
    ofstream output;
    //open the file
    output.open("Times.dat");
    //declare and initialize variables
    const int TOTCHNS1=12;//total chance of the level 1 and 2
    const int TOTCHNS2=18;//total chance of the level 3
    const int SIZE1=4;//smaller size
    const int SIZE2=6;//larger size
    string dash;
    string answer;
    int size;
```

```
   int level;
   int tyTime=0;  //how many tyTime user tried
   int strTime;   //start time of the game
   int endTime;   //ending time of the game
   int usdTime;   //total time player play the game
   int gusCorr=0;//how many correct number have been guessed
   int chnsLft;  //chance counter(how many chances left)
   int hin=0;//count how many times player use hint
   int digit=0;  //digit of the user guesses
   char guess=0; //the number user guesses
   char pswd[SIZE2]={};//the password store in the array
   vector<int> inputDg;//the digits finished

   //introduce the game
   introduce();
   //Prompt user for level
   do {
      cout<<endl<<"*****************LEVEL
SELECTION*****************"<<endl;
      cout<<"    1.Easy:"<<endl;
      cout<<"      Feature: (1)4 digit"<<endl;
      cout<<"               (2)No Repeat Number"<<endl;
      cout<<"               (3)Including sample input"<<endl;
      cout<<"    2.Normal:"<<endl;
      cout<<"      Feature: (1)4 digit"<<endl;
      cout<<"               (2)Might has repeat number"<<endl;
      cout<<"               (3)Including a hint for a digit"<<endl;
```

```
    cout<<"    3.Hard"<<endl;
    cout<<"      Feature: (1)6-digit"<<endl;
    cout<<"               (2)Might has repeat number"<<endl;
    cout<<"               (3)Including a hint for a digit"<<endl;
    cout<<"Please choose a level(Type 1,2,or 3)"<<endl;
    cin>>level;
    if(level<1||level>3)
        cout<<"Invalid Input"<<endl;
}while(level<1||level>3);
//According to level that player chose, set the size of pswd and chances left
if(level==1||level==2) {
    size=SIZE1;
    chnsLft=TOTCHNS1;
} else {
    size=SIZE2;
    chnsLft=TOTCHNS2;
}
//get a random password and put it in array
gtPswd(pswd,level,SIZE1,SIZE2);
//Use for loop get the password into strings
for(int i=0;i<size;i++) {
    answer+=pswd[i];
}
dash=toDash(size);//get the dash
strTime=time(0);//record beginning time
//game begins
while(chnsLft>0&&gusCorr<size) {
```

```
    //Prompt user for the guess
    cout<<endl;
    cout<<"The password now looks like this: "<<dash<<endl;//Output dash
    cout<<"You have "<<chnsLft<<" chances left"<<endl;//output chances left
    ask(guess,digit,size,level,inputDg,pswd,hin);//Prompt user for guess
    char result=check(guess,digit,pswd,size);//check the guess
    switch(result) {
       case'1': { //if the number and place both are correct
          if(inside(inputDg,digit)) { //if user have finished that digit
             cout<<"You already finished this digit,"
                <<"try other digits"<<endl;
          } else {//user didn't finish this digit
             replace(dash,guess,digit);//replace of the correct digit
             inputDg.push_back(digit-1);//record the digit which has been finished to
vector
             cout<<"Your guess is correct."<<endl;
             gusCorr++;
          }
          break;
       }
       case'2': { //if number is correct but digit is wrong
          cout<<"This is the correct number but in wrong place."<<endl;
          chnsLft--;
          break;
       }
       case'3': { //if number and digit both wrong
          cout<<"Wrong number and wrong place."<<endl;
```

```
            chnsLft--;
            break;
         }
         default:;
      }
      tyTime++;//keep track of how many tyTime user have input
   }
   endTime=time(0);
   usdTime=endTime-strTime;//Get the used time
   if(chnsLft==0) { //No chances left for player
      cout<<"You lost"<<endl;
      output<<"You lost"<<endl;
   }
   if(gusCorr==size) { //when 4 digits have been guessed correctly
      output<<"You win this game after "<<tyTime<<" tries"<<endl;
      cout<<"You win this game after "<<tyTime<<" tries"<<endl;
      record(usdTime);
   }
   cout<<endl<<"The answer is "<<answer<<endl;
   cout<<"You spent "<<usdTime<<" seconds to play this game"<<endl<<endl;
   dspRecd();//display the record
   output.close();

   //Exit stage right
   return 0;
}
//According to level,generate the dash that will be displayed
```

```
string toDash(int size) {
   string dashed="";
   for(int i=0;i<size;i++) {
      dashed+="-";
   }
   return dashed;
}
//Introduction and Rule of the game
void introduce() {
   cout<<"****************** Welcome to Bomb password
******************"<<endl;
   cout<<"*  In this game, you should guess the 4 or 6-digit password   *"<<endl;
   cout<<"*     First, you will input only one number that you guess    *"<<endl;
   cout<<"*         Then, you will input the digit of this number       *"<<endl;
   cout<<"*    The digit of the number from left to right is 1,2,3...   *"<<endl;
   cout<<"*      After you input these two information,the computer     *"<<endl;
   cout<<"*    will tell you whether the number and digit are correct   *"<<endl;
   cout<<"* If your guess is correct, your chances left won't count down*"<<endl;

cout<<"************************************************************"
<<endl;
   cout<<"Press Enter to start the game";
   cin.ignore();
}
//ask user for guessing
void ask(char& guess,int& digit,int size,int level,vector <int>ipDg,char pswd[],int &hin)
{
   do {
```

```
    cout<<"Please input a 1-digit number you guess(0-9)"<<endl;
    if(level==1)
       cout<<"If you need sample for input, type \'s\'"<<endl;
    if((level==2||level==3)&&hin<1)
       cout<<"If you need hint for answer, type \'h\'"<<endl;
    cin>>guess;
    cin.ignore();
    if((guess=='s'||guess=='S')&&level==1) { //when player need sample
       sample();//output sample via ifstream
    } else if((guess=='h'||guess=='H')&&(level==2||level==3)&&hin<1) {
       //when player needs hint
       //limit only 1 hint player can get
       hint(ipDg,pswd,size);//display the hint
       hin++;
    } else if(guess<48||guess>57) {
       cout<<"Invalid input"<<endl<<endl;
    }
  } while(guess<48||guess>57);
  do {
    cout<<"Please input the digit of this number"<<"(1-"<<size<<")"<<endl;
    cin>>digit;
    cin.ignore();
    if(digit<1||digit>size)
       cout<<"Invalid input"<<endl<<endl;
  } while(digit<1||digit>size);
}
```

```
char check(char guess,int digit,const char pswd[],int size) {
   //Because array counts from 0,but digit from left to right is 1,2,3,4,so it need digit-1
for array
   if(guess==pswd[digit-1]) { //when guess and digit are correct
      return '1';
   } else if(indexOf(guess,pswd,size)) { //number is right but digit is wrong
      return '2';
   } else { //both are wrong
      return '3';
   }
}
//return whether the char is in the array
bool indexOf(char x,const char pswd[],int size) {
   bool temp=false;
   for(int i=0;i<size;i++) {
      if(pswd[i]==x)
         temp=true;
   }
   return temp;
}

void replace(string& dash,char guess,int digit) {
   string part1=dash.substr(0,(digit-1));
   string part2=dash.substr(digit);
   dash=part1+guess+part2;
}
//return whether this digit has been finished
```

```
bool inside(const vector<int> inputDg,int digit) {
    bool temp=false;
    for(int i=0;i<inputDg.size();i++) {
        if(inputDg[i]==(digit-1))
            temp=true;
    }
    return temp;
}
//display sample via file
void sample() {
    string str;
    ifstream input;
    input.open("Sample.dat");
    while(input>>str) {
        if(str=="Now"||str=="If"||str=="You"||str=="<"||str=="Then,")
            cout<<endl;
        cout<<str<<' ';
    }
    cout<<endl<<endl;
    input.close();
}
//randomly generate the password according to level
void gtPswd(char pswd[],int level,const int SIZE1,const int SIZE2){
    bool diff=true;
    if(level==1) { //every digit is different number
        for(int i=0;i<SIZE1;i++) {
            char temp;
```

```
            do {
                diff=true;
                temp=rand()%10+'0';
                for(int j=0;j<i;j++) {
                if(temp==pswd[j]) diff=false;
                }
            }while(diff==false);
            pswd[i]=temp;
        }
    } else if(level==2) {
        for(int i=0;i<SIZE1;i++) {
            pswd[i]=rand()%10+'0';
        }
    } else {
        for(int i=0;i<SIZE2;i++) {
            pswd[i]=rand()%10+'0';
        }
    }
}
//Display the hint for player
void hint(vector<int> ipDg,char pswd[],int size) {
    int temp=0;
    int tip=0;
    bool done=false;//determine whether this digit has been done
    if(ipDg.size()!=0) { //when player already finish some digits
        do {
            done=false;
```

```
            temp=rand()%size;
            for(int i=0;i<ipDg.size();i++) {
                if(temp==ipDg[i]) done=true;
            }
        }while(done);
    } else { //when player didn't finish any digits
        temp=rand()%size;
    }
    //change the tip from char to integer
    tip=static_cast<int>(pswd[temp]-48);
    //Display the hint in binary form
    cout<<"Digit "<<(temp+1)<< " in binary is "<<bin(tip)<<endl;
}
//change the decimal to binary
int bin(int a) {
    int rm;//remainder
    int rs;//result
    int temp=1;
    vector<int> ary;//to store each digit of binary number
    while(a>0) {
        rm=a%2;
        a/=2;
        ary.push_back(rm);
    }
    for(int i=0;i<ary.size();i++) {
        rs+=temp*ary[i];
        temp*=10;
```

```
    }
    return rs;
}
//Record the playing time and sort the time record
void record(int usdTime) {
    vector<int> tmRecd;
    int tm;
    //Get the previous records from file
    ifstream input;
    input.open("Record.dat");
    while(input>>tm) {
        tmRecd.push_back(tm);
    }
    input.close();
    //Save the playing time to vector
    tmRecd.push_back(usdTime);
    //Sort the record with vector
    for(int i=0;i<tmRecd.size()-1;i++) {
        for(int j=i+1;j<tmRecd.size();j++) {
            if(tmRecd[i]>tmRecd[j]) {
                int temp=tmRecd[i];
                tmRecd[i]=tmRecd[j];
                tmRecd[j]=temp;
            }
        }
    }
    //Output the current record to file
```

```
    ofstream output;
    output.open("Record.dat");
    for(int i=0;i<tmRecd.size();i++) {
        output<<tmRecd[i]<<endl;
    }
    output.close();
}
//Get the Top 10 Records and display
void dspRecd() {
    //declare variables
    const int ROW=15;
    int tm;//the time record from file
    int rc[ROW][COL]={};//2D array
    //Open the file
    ifstream input;
    input.open("Record.dat");
    for(int i=0;i<10&&input>>tm;i++) {
        rc[i][0]=(i+1);
        rc[i][1]=tm;
    }
    input.close(); //Close the file
    cout<<"*********Top 10 Records*********"<<endl;
    cout<<"  Rank       Time(seconds)"<<endl;
    //Formatted output the Top 10 records
    for(int i=0;i<10;i++) {
        for(int j=0;j<COL;j++) {
            if(rc[i][j]!=0)
```

```
            cout<<setw(5)<<rc[i][j]<<"          ";
        else
            cout<<"  ----         ";
    }
    cout<<endl;
  }
}
```