Project 1

<Guessing Password>

Name: Haolan Ye (Benjamin)

Class: CSC-5 40717

Date: 01/31/2015

1. Inspiration:

This game is to guessing a 4-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 pass-word. 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. First, computer will generate a 4-digit number (from 0000 to 9999). Then, players only can guess one digit of password at one time. Players should input the 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 chances to guess the password. When the player guesses the number correctly, the chances won't change. If players cannot guess all digits of password after 12 chances, the computer will display game over.

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. Players cannot guess the same number in a specific digit after this digit has been done. Every time players input the number, they only can input a 1-digit number or "s"(for sample input).

3. Summary:

Total Line of Code	200
Blank Line	11
Comment Line	25
Variable	13
Function	8

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. Approximately, I spent 7 hours to code and check this game (not including documentation). 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.

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.

5. Pseudo Code

Introduce the game

Randomly get a 4-digit password, store the copy of the password in string

Create dash (string)

When (chance left>0 and the number of correct guesses<4)

Display the dash and chances left

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

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

Output the result of this game and output to a file

6. System Libraries

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

7. Variables List

Type	Variable Name	Description	Declare Location(line)
ofstream	output	Output to the file	35
const int	TOTCHNS	Total chance of game	39
	SIZE	Size of the password	40
string	dash	Password look like	41
	answer	Answer of password	42
	dashed	String that returns to dash	114
	part1	Substring from the first digit to one digit before guess	175
	part2	Substring from one digit after guess to the end of password	176
	str	Store the string that comes from file	190
int	times	Times player tried	43
	gusCorr	Number of correct guess	44
	chnsLft	How many chance left	45
	digit	The digit of number that player guess	46
char	guess	The number player guess	47
	pswd[SIZE]	The password that store in char array	48
	result	The result of one guess	70
vector<int>	inputDg	Done digits	49
ifstream	input	Input the file to display onto the console	191
boolean	temp	Temp for boolean	166,181

8. Function

Type	Name	Argument	Function	Location
string	toDash	int	Get the dash for display	113
void	introduce	None	Introduce the game	120
void	ask	char&,int&	Prompt player for number and digit	134
char	check	char, const char[],int	check whether number and digit are correct	154
bool	indexOf	char, const char[],int	return whether the char is in the char array	165
void	replace	string&,char,int	replace of the correct digit	174
bool	inside	const vector<int>,int	return whether this digit is finished	180
void	sample	None	display the sample of guessing	189

9. Concept covered

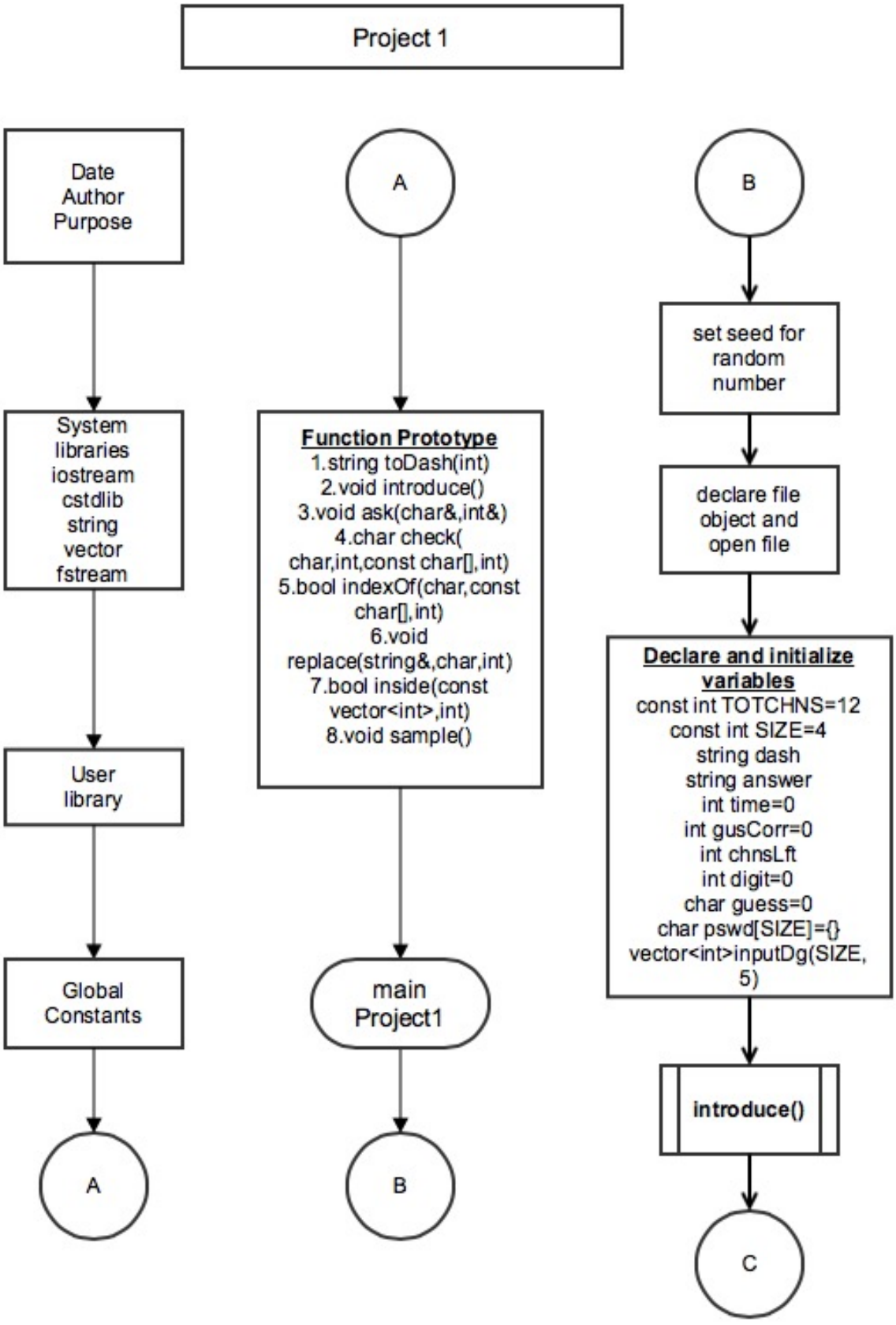
Concept	Type	Code	Location(line)
Input / Output	cout	cout<<endl	66
	cin	cin>>guess	138
Variables	int, char,string, ifstream,ofsreamt	int times=0	43
Comment		//system Libraries	8
Type casting	static_cast<type>	static_cast<unsigned short>(time(0))	33

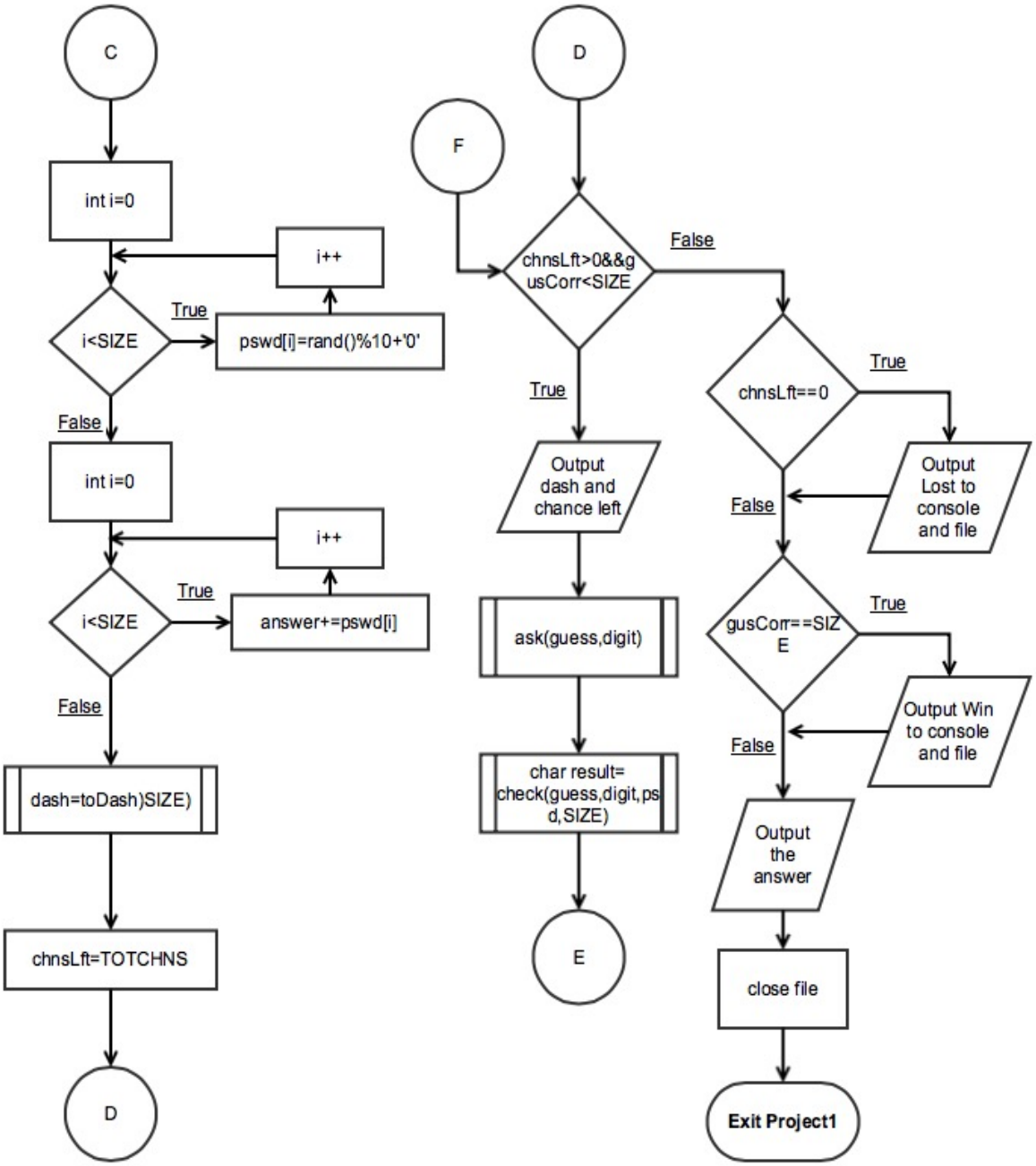
Making decisions	if	if(chnsLft==0)	98
	if-else	if(inside(inputDg,digit)) else	74

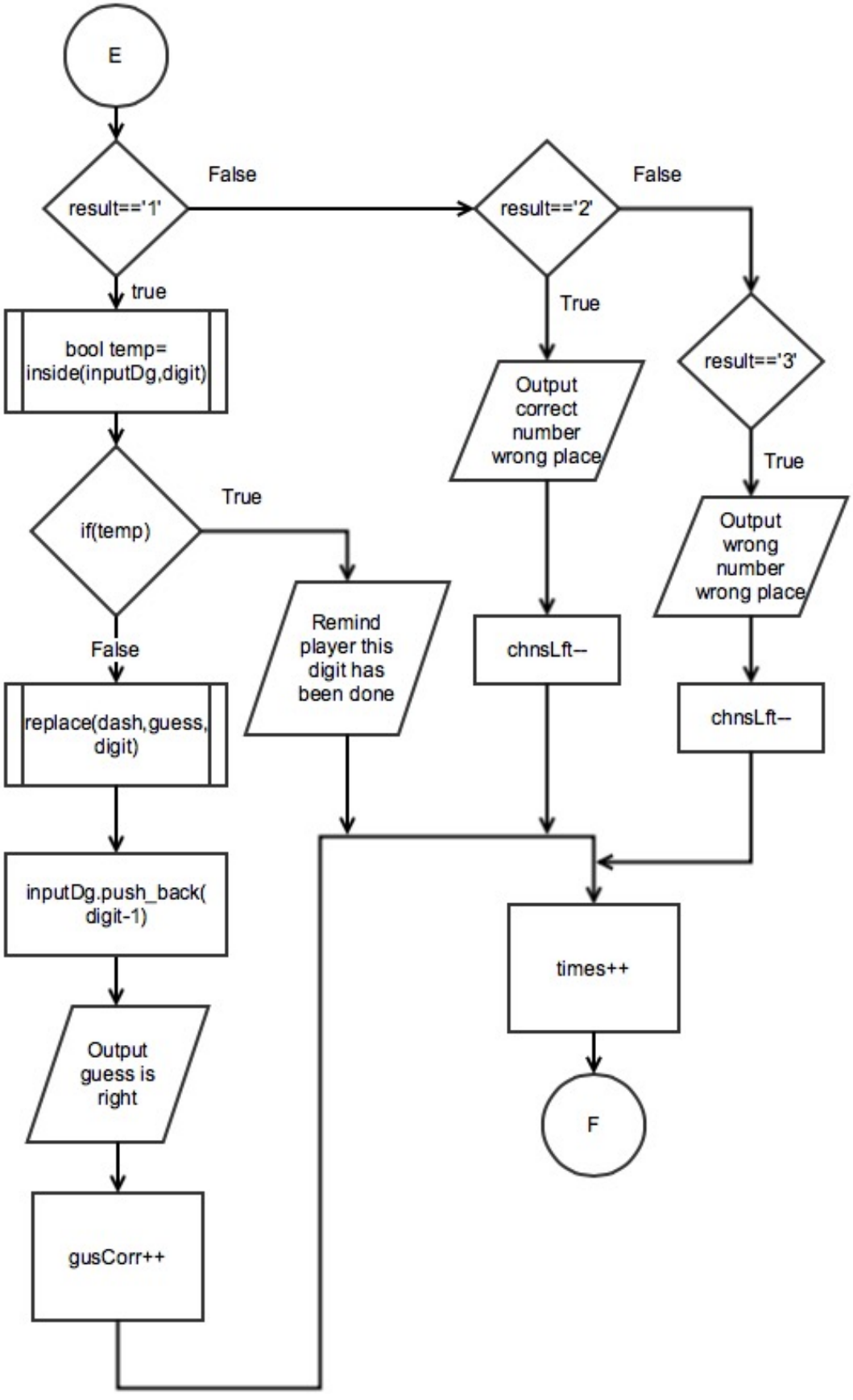
	if-else-if	if(guess==pswd[digit-1])
		else if(indexOf(guess,pswd,size))
		else
			156

	switch	switch(result)	71
Loop	for	for(int i=0;i<SIZE;i++)	54
	while	while(chnsLft>0&&gusCorr<SIZE)	64
	do-while	do {} while (guess<48||guess>57)	135
File I/O	ifstream	ifstream input; input.open("Sample.dat");	191
	ofstream	ofstream output; output.open("Times.dat");	35
Function	void, int, string, char, bool	string toDash(int)	21
Array	char[]	char pswd[SIZE]={}	48
vector	vector<int>	vector<int> inputDg(SIZE,5)	49

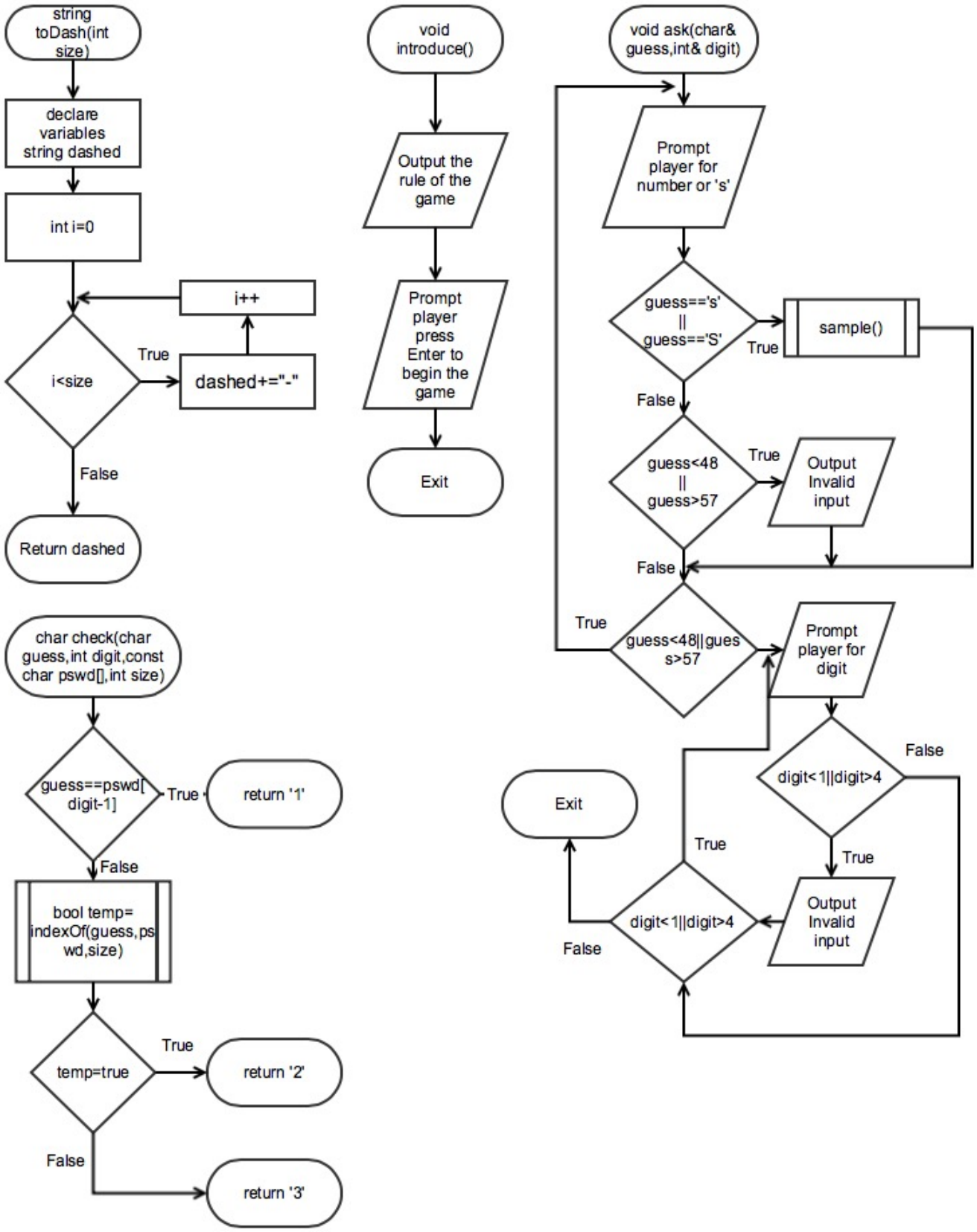
10. Flowchart (1)main flowchart(3 pages)

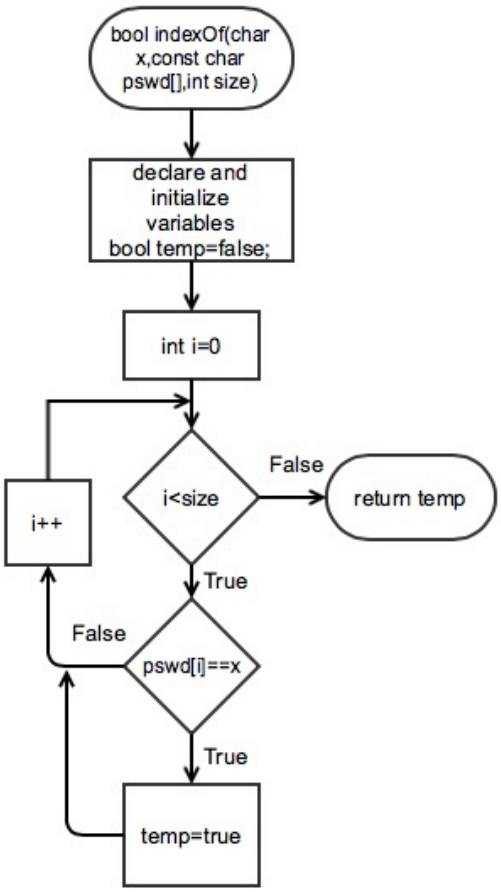


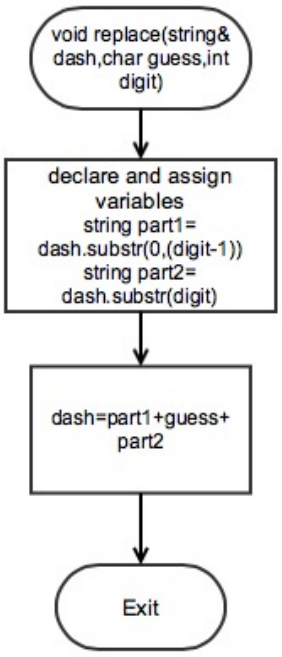

C
int i=0
i++
i<SIZE
True
pswd[i]=rand()%10+'0'
False
int i=0
i++
i<SIZE
True
answer+=pswd[i]
False
dash=toDash)SIZE)
chnsLft=TOTCHNS
D
F
D
chnsLft>0&&g
usCorr<SIZE
False
True
Output
dash and
chance left
ask(guess,digit)
char result=
check(guess,digit,ps
d,SIZE)
E
chnsLft==0
True
Output
Lost to
console
and file
False
gusCorr==SIZ
E
True
Output Win
to console
and file
False
Output
the
answer
close file
Exit Project1

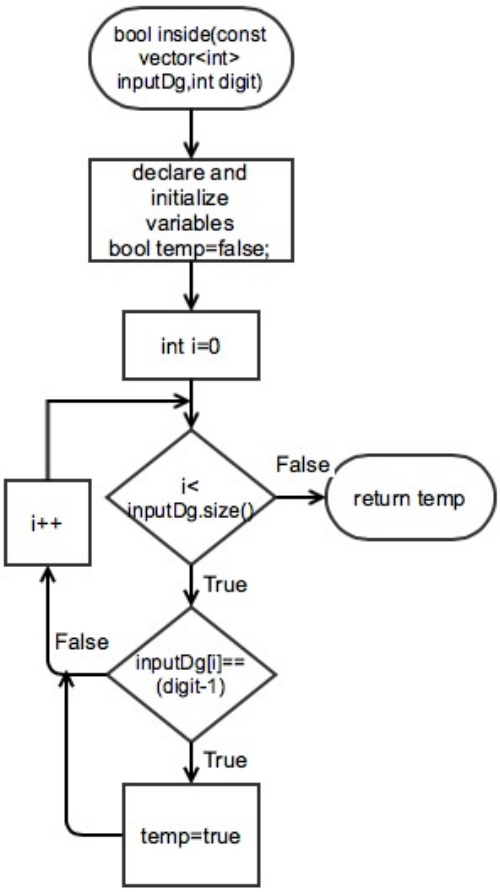

E
result=='1'
False
result=='2'
False
true
True
result=='3'
bool temp=
inside(inputDg,digit)
Output
correct
number
wrong place
True
Output
wrong
number
wrong place
if(temp)
True
Remind
player this
digit has
been done
False
chnsLft--
chnsLft--
replace(dash,guess,
digit)
inputDg.push_back(
digit-1)
times++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

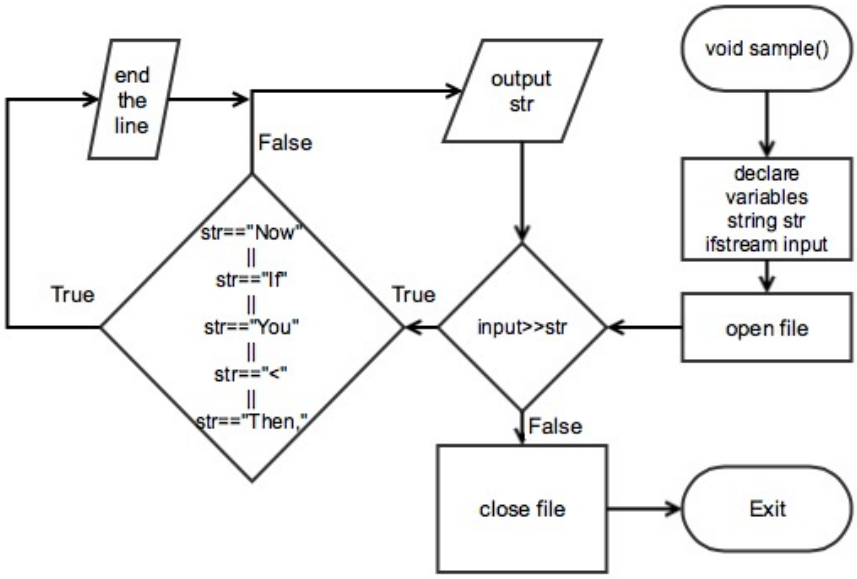
(2) Function flowchart(2 pages)




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
close file
Exit
True
str=="Now" || str=="If" || str=="You" || str=="<" || str=="Then,"
True
False
output str
end the line

Code:

```
/*
 * File:   main.cpp
 * Author: Haolan Ye(Benjamin)
 * Created on January 27, 2015, 12:32 PM
 * Purpose: Project1(Name of the game:Bomb Password)
 */

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

//User Libraries

//Global Constants

//Function Prototypes
string toDash(int);//change the password to dash
void introduce();//introduce the game
void ask(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

//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 TOTCHNS=12;//total chance of the game
    const int SIZE=4;//the size of the char array
    string dash;
    string answer;
    int times=0;  //how many times user tried
    int gusCorr=0;//how many correct number have been guessed
    int chnsLft;  //chance counter(how many chances left)
    int digit=0;  //digit of the user guesses
    char guess=0; //the number user guesses
    char pswd[SIZE]={};//the password store in the array
    vector<int> inputDg(SIZE,5);//the digits finished

    //introduce the game
    introduce();
```

```
    //get a random 4-digit password and put it in array
    for(int i=0;i<SIZE;i++) {
        pswd[i]=rand()%10+'0';
    }
    //Use for loop get the password into strings
    for(int i=0;i<SIZE;i++) {
        answer+=pswd[i];
    }
    dash=toDash(SIZE);//get the dash
    chnsLft=TOTCHNS;
    //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);//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 finish 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:;
  }
  times++;//keep track of how many times user have input
}
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 "<<times<<" tries"<<endl;
  cout<<"You win this game after "<<times<<" tries"<<endl;
}
```

```
    cout<<"The answer is "<<answer<<endl;
    output.close();

    //Exit stage right
    return 0;
}

string toDash(int size) {
    string dashed="";
    for(int i=0;i<size;i++) {
        dashed+="-";
    }
    return dashed;
}
void introduce() {
    cout<<"****************** Welcome to Bomb password
******************"<<endl;
    cout<<"*    In this game, you should guess the 4-digit password   *"<<endl;
    cout<<"*       First, you will input a 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 1,2,3,4   *"<<endl;
    cout<<"*    After you input these two information,the computer    *"<<endl;
    cout<<"*  will tell you whether the number and digit are correct  *"<<endl;
    cout<<"* Attention: some digits of password may be the same number*"<<endl;

cout<<"************************************************************"<<e
ndl;
    cout<<"Press Enter to start the game";
```

```
    cin.ignore();
}

void ask(char& guess,int& digit) {
    do {
        cout<<"Please input a number you guess"<<endl;
        cout<<"If you need sample for input, type \'s\'"<<endl;
        cin>>guess;
        cin.ignore();
        if(guess=='s'||guess=='S') //when player need sample
            sample();              //output sample via ifstream
        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"<<endl;
        cin>>digit;
        cin.ignore();
        if(digit<1||digit>4)
            cout<<"Invalid input"<<endl<<endl;
    } while(digit<1||digit>4);
}

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 is 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();
}
```