Comp 1011 Individual project report(topic 1)

1.Description of the project:

Problem statement: we defined the problem as to design a system for the administer of a bank to manage the personal information of users.

2.The objective of the program:

To allow the user to add information, and after they add the information, they cannot edit it, and it support the administer to check the information using block number, ID number, and the hash value. Also can let them to check the integrity of each blocks to ensure the safety issue.

Project design:

(a) structure

```
struct identity{
    char user_name[100];
    char ID_number[100];
    char age[100];
    char bankaccount[100];
    char bankpassword[90];
    identity *next=NULL;
    char hash_value[41];
};//to make a struct
```

Structure

The unit to store information, it can also store the hash value of the previous block, so the first one do not store any hash value, also the last block only stored the hash value. That is a special case we need to consider in the following function design.

First, I define the structure identity,which is one unit if the personal information, it includes the username, ID number, age, bank account and password, also the hash value of the previous block. The pointer next can be used to link the blocks together

```
void menu();
void check_menu();
char* hash_string(char*,char*);
char* hash_identity(identity*);
struct identity* add_information(identity*);
int check_information(identity*);
int check_with_number(identity*);
int check_with_hash_value(identity*);
int check_with_ID_number(identity*);
void print_information(identity*);
int check_integrity(identity*);
```

Photo-types

To ignore the order of each function, make it much more convenient to call them. Also from the photo-types we can clearly know which function can it achieve, and directly find the codes.

I have defined 11 functions to achieve different part's function.

(b)menu

```
72
73  void menu(){
74      cout<<"************************************"<<endl;
75      cout<<"*Welcome to bankaccount saving center*"<<endl;
76      cout<<"************************************"<<endl;
77      cout<<"*      please select ur choice:      *"<<endl;
78      cout<<"*      0.u want to quit             *"<<endl;
79      cout<<"*      1.add ur information         *"<<endl;
80      cout<<"*      2.check information          *"<<endl;
81      cout<<"*      3.check the integrity        *"<<endl;
82      cout<<"*      4.clean the screen           *"<<endl;
83      cout<<"************************************"<<endl;
84  }//output the menu for the main function
```

Here is the main menu, which tells the user to enter 0 to quit the system, 1 to add their information, 2. to check their information, 3. to check the integrity of the blocks,it allows them to check whether someone have invade the system and change the data. 4. to clean the screen, make ensure no one can view the data they have input after they finished using.

(c)main function

```
int main() {
    identity* head=new identity;
    head->next=NULL;
    identity* first=head;
    int choice;
    while(true){
        menu();
        cin>>choice;
        cin.ignore();
        switch(choice){
            case 0:
                cout<<"Thanks for using our bank account center, have a nice day ^_^"<<endl;
                exit(0);
            case 1:
                first=add_information(first);
                system("pause");
                break;
            case 2:
                check_information(head);
                break;
            case 3:
                check_integrity(head);
                system("pause");
                break;
            case 4:
                system("cls");
                break;
            default:
                cout<<"WRONG MESSAGE Please try again"<<endl;
        }
    }
}
```

main function

The pointer head is going to point at the first node, we cannot move it among the linked list, because for each function, check, add, search the information, we need to start from the pointer head. So each time when we need to use it, we need to copy it.

I have design a head pointer to point to the first block, so that in each function, I can load data from the first block. Then ask the use a switch case structure to let the user to choose the function they want to use, and call different functions to achieve the function.

(d)introduction to different function

- When the user enter 0, as the exit been called, they can quit the program.
- When the user enter 1, as the function add_information(), the system will ask them to add their information. The pointer first copy from the pointer head is aim to move along the linked list. In order to add new nodes.

```
identity* add_information(identity* head){
    identity* newpoint=new identity;
    head->next=newpoint;
    cout<<"please enter ur username: "<<endl;
    cin.getline(head->user_name,41);
    cout<<"please enter ur ID: "<<endl;
    cin.getline(head->ID_number,100);
    cout<<"Please enter ur age: "<<endl;
    cin.getline(head->age,100);
    cout<<"Please enter ur account: "<<endl;
    cin.getline(head->bankaccount,100);
    cout<<"please enter ur password:"<<endl;
    cin.getline(head->bankpassword,90);
    strncpy(newpoint->hash_value,hash_identity(head),41);
    cout<<"the hash value is: "<<newpoint->hash_value<<"(please remember it and keep secret)"<<endl;
    head=head->next;
    return head;
}//to input the information
```

Function Add_information()

Let the current node to store the information, and store the hash value into the next node of the linked list. Finally update the current node to be the next node(can also say move the pointer).

Each time when the function been called, it will store the information that the user have input, then call the function hash_identity() to hash the information have already been stored in the current node, then automatically create a new node, and link it with the current node to store the result return by hash_identity() in the hash_value part.

```
char* hash_identity(identity *point){
    char dest[531];
    strcpy(dest,point->user_name);
    strcat(dest,point->ID_number);
    strcat(dest,point->age);
    strcat(dest,point->bankaccount);
    strcat(dest,point->bankpassword);
    strcat(dest,point->hash_value);
    char hash[41];
    char* m=hash_string(dest,hash);
    return m;
}//to hash a whole block
```

Function hash_identity()

First copy the username in to the array named dest, then append the other information after it, last call the function provided in the source file, pass dest into the function to get the result m, last return the final result.

- When the user enter 2, as the function check information been called, it will display the check menu to guide users to check the information inside of the blocks by using different ways.

```
int check_information(identity* head){
    if(head->next==NULL){
        cout<<"you haven't input any blocks yet"<<endl;
        system("pause");
        return 0;
    }
    check_menu();
    while(true){
        int condition;
        cin>>condition;
        switch(condition){
            case 1:
                check_with_number(head);
                break;
            case 2:
                check_with_hash_value(head);
                break;
            case 3:
                check_with_ID_number(head);
                break;
            default:
                system("pause");
                return 0;
        }
        check_menu();
    }
}//to order three choices for the user to check the information
```

Function check_information():

Can allow the users to choose different ways to check the information stored in different blocks. It will first check the head->next, to see whether the user have already input something or not. If it is NULL, the system will ask them to input something into the block first. Otherwise, the user can input the number of the choice according to the check menu to use different ways to check the information inside of different blocks.

```
void check_menu(){
    cout<<"*******************************"<<endl;
    cout<<"*    please select the way     *"<<endl;
    cout<<"*******************************"<<endl;
    cout<<"*1.check with block number     *"<<endl;
    cout<<"*2.check with hash value       *"<<endl;
    cout<<"*3.check with ID_number        *"<<endl;
    cout<<"*4.click any other key to quit*"<<endl;
    cout<<"*******************************"<<endl;
}//to display the choices
```

❖ Enter 1 to check with the block number, by using this function, the user can enter the number of the block to get the information stored in the bock.

```
int check_with_number(identity* head){
    int block_number;
    cout<<"please enter the block number(start from 0)"<<endl;
    cin>>block_number;
    int num=0;
    identity* monitor=new identity;
    monitor=head;
    while((monitor->next!=NULL)&&(num<block_number)){
        monitor=monitor->next;
        num++;
    }
    if((monitor->next==NULL)||(block_number<0)){
        cout<<"out of range"<<endl;
        system("pause");
        return 0;
    }
    else{
        print_information(monitor);
    }
}//to let the user to check with the lock number
```

Function check_with_number()

First we ask the user to enter the number of the block they want to check. Then just increase the number of the counter, at the same time, move the pointer until we get the target block. Then output the information. (here we won't output the hash value to ensure the safety issue),it can also deal with the out of range problem.

❖ Enter 2 to check with the hash value, the user need to first remember the hash value of the target block, then input it to login the block they want to view.

```
int check_with_hash_value(identity* head){
    char hash[41];
    cout<<"please enter the hash value:"<<endl;
    cin>>hash;
    cin.ignore();
    identity* monitor=new identity;
    monitor=head;
    while((strcmp((monitor->next)->hash_value,hash)!=0)&&((monitor->next)->next!=NULL)){
        if((monitor->next)->next!=NULL){
            monitor=monitor->next;
        }
    }
    if(strcmp((monitor->next)->hash_value,hash)!=0){
        cout<<"no such block!!!"<<endl;
        system("pause");
        return 0;
    }
    if(strcmp((monitor->next)->hash_value,hash)==0){
        print_information(monitor);
    }
}//check with hash value
```

Function check_with_hash()

In this function, I need to compare the hash value stored in the next node with the input of the user. Until the next pointer inside of the next node is NULL(the special case mentioned in the main function part), I will stop the compare and tell the user it is not exist, otherwise output the information of the target block.

❖ Enter 3 to check with the ID number, the user need to input the ID number stored in the target block they want to check to get the information.

```
int check_with_ID_number(identity* head){
    char num[100];
    cout<<"please enter the ID number:"<<endl;
    cin>>num;
    identity* monitor=new identity;
    monitor=head;
    while((strcmp(monitor->ID_number,num)!=0)&&(monitor->next!=NULL)){
        monitor=monitor->next;
    }
    if(monitor->next==NULL){
        cout<<"No such an ID!!!"<<endl;
        system("pause");
        return 0;
    }
    if(strcmp(monitor->ID_number,num)==0){
        print_information(monitor);
    }
}//check with ID number
```

Function check_with_ID_number()

In this function, I need to compare the user's input with each ID stored in the blocks, until it is same, or it reaches the end of the linked list. If it reaches the end, we tell the user no such a block. Otherwise the system will output the information stored in the target block to the users.

- Enter 3 to choose the third function, to check the the integrity if the whole block chain, it can ensure the safety issue.

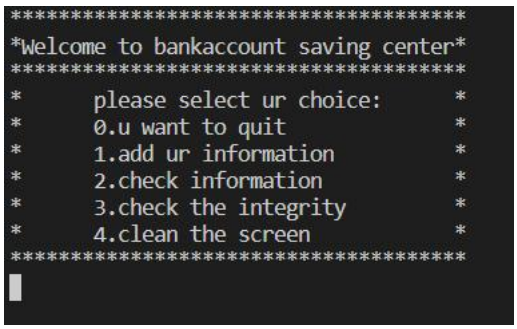
```
int check_integrity(identity *head){
    if(head->next==NULL){
        cout<<"please add some block first"<<endl;
        return 0;
    }
    identity *monitor=new identity;
    monitor=head;
    int num=0;
    while((monitor->next)->next!=NULL){
        if(strcmp(hash_identity(monitor),(monitor->next)->hash_value)!=0){
            cout<<"it have been invaded at"<<num<<"block"<<endl;
            return 0;
        }
        monitor=monitor->next;
        num++;
    }
    cout<<"No thing happened, fine."<<endl;
}
```

Function check_integrity()

It will first check the next pointer to see whether it is empty or not, if so it will stop and remind the user. We will create a new node to point to the head node, then compute the hash value of each block, compare it with the hash value stored in the next block, at the same time move the pointer. If it is different the system will raise a warning.

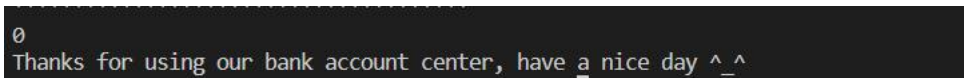
- Enter 4 to clean the screen if the user don't want the next person to see their manipulate.

User manual

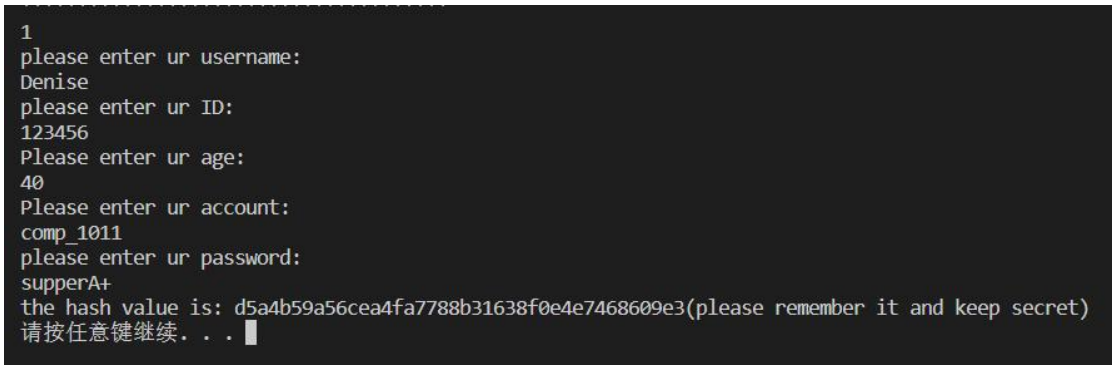


When you run the program u can see the menu of the main function.

- Enter 0 to quit the system



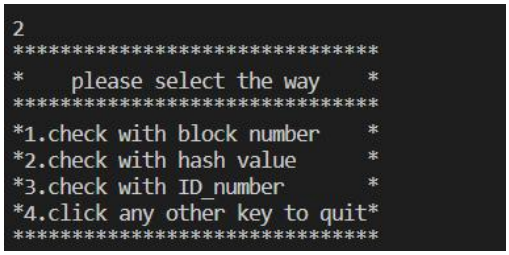
- Enter 1 to add the information into the system



As it display here, it will let you to remember the hash value.

- Enter 2 to check the information

It will first display the menu to let you know the ways to login to view your target block.



If you haven't add any information into the system.you will get a warning.

❖ Next you can enter one to check with the block number.

```
1
please enter the block number(start from 0)
0
user name:      Denise
ID number:      123456
age:            40
bank account:   comp_1011
password:       supperA+
请按任意键继续. . .
```

For example, as it display in this screen shot, you can view the 0th information.

Otherwise if the block doesn't exist it will show you the following message.

```
1
please enter the block number(start from 0)
-1
out of range
```

- You can enter 2 to check with the hash value as it shows in the following picture.

```
2
please enter the hash value:
d5a4b59a56cea4fa7788b31638f0e4e7468609e3
user name:      Denise
ID number:      123456
age:            40
bank account:   comp_1011
password:       supperA+
请按任意键继续. . .
```

If the hash value you have passed is not exist, it will also give some warning message.

```
2
please enter the hash value:
d5a4b59a56cea4fa7788b31638f0e4e746860213
no such block!!!
请按任意键继续. . .
```

- You can enter 3 to check with the ID number which have been stored in the block. As it shows in the following picture.

```
please enter the ID number:
123456
user name:      Denise
ID number:      123456
age:            40
bank account:   comp_1011
password:       supperA+
请按任意键继续. . .
```

If it not exist, the system will display some warning message.

```
please enter the ID number:
12
No such an ID!!!
请按任意键继续. . .
```

- You can enter 4 to quit the check process.

- You can enter 3 to check the integrity of the whole linked list. As it display in the following picture.

If you haven't add any information into the system.you will get a warning.

- You can enter 4 to clean the screen make sure no one can see your manipulate.

This is the whole guide of the system, have a good journey with us.