Full name: Nguyen Xuan Chu

Student code: 240053

Table of Contents

Mini Project C

Solution of question 1:

```
#include<stdio.h>
#include<string.h>

typedef struct
{
    int id;
    char name[30];
    float monthly_salary;
    float yearly_salary;
} Employee;
float calc_yearly_salary(float monthly_salary)
{
    return monthly_salary*12;
}

int main()
{
    int n;
    printf("Enter the number of employees: ");
    scanf("%d", &n);

    Employee e[n];
    char ch[30];

    for (int i=0; i<n; i++)
    {
        printf("### Enter Employee Infomation ###\n");
        printf("Enter ID: ");
        scanf("%d", &e[i].id);
```

```
        printf("Enter name: ");
        scanf("%s", ch);
        strcpy(e[i].name, ch);
        printf("Monthly Salary: ");
        scanf("%f",&e[i].monthly_salary);
        e[i].yearly_salary = calc_yearly_salary(e[i].monthly_salary);
        printf("\n");
    }
    printf("=== === All Employee's Details === ===\n");
    for (int i=0; i<n; i++)
    {
        printf("Employee number %d:\n", i+1);
        printf("ID: %d\n", e[i].id);
        printf("Name: %s\n", e[i].name);
        printf("Monthly Salary: %f\n", e[i].monthly_salary);
        printf("Yearly Salary: %f\n", e[i].yearly_salary);
        printf("\n");
    }
    return 0;
}
```

Result of question 1:

```
nxc@nxc-virtual-machine:~$ cd Documents/MiniProject/
nxc@nxc-virtual-machine:~/Documents/MiniProject$ gcc q1.c
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter the number of employees: 2
### Enter Employee Infomation ###
Enter ID: 1
Enter name: nguyen
Monthly Salary: 100

### Enter Employee Infomation ###
Enter ID: 2
Enter name: xuan
Monthly Salary: 200

=== === All Employee's Details === ===
Employee number 1:
ID: 1
Name: nguyen
Monthly Salary: 100.000000
Yearly Salary: 1200.000000

Employee number 2:
ID: 2
Name: xuan
Monthly Salary: 200.000000
Yearly Salary: 2400.000000

nxc@nxc-virtual-machine:~/Documents/MiniProject$
```

Solution of question 2:

```
#include<stdio.h>

int len(char *ptr)

{

    int count = 0;

    while((*ptr) != '\0')

    {

        ++count;

        ptr++;

    }

    return count;

}

void concatenate(char *pd, char *ps)
```

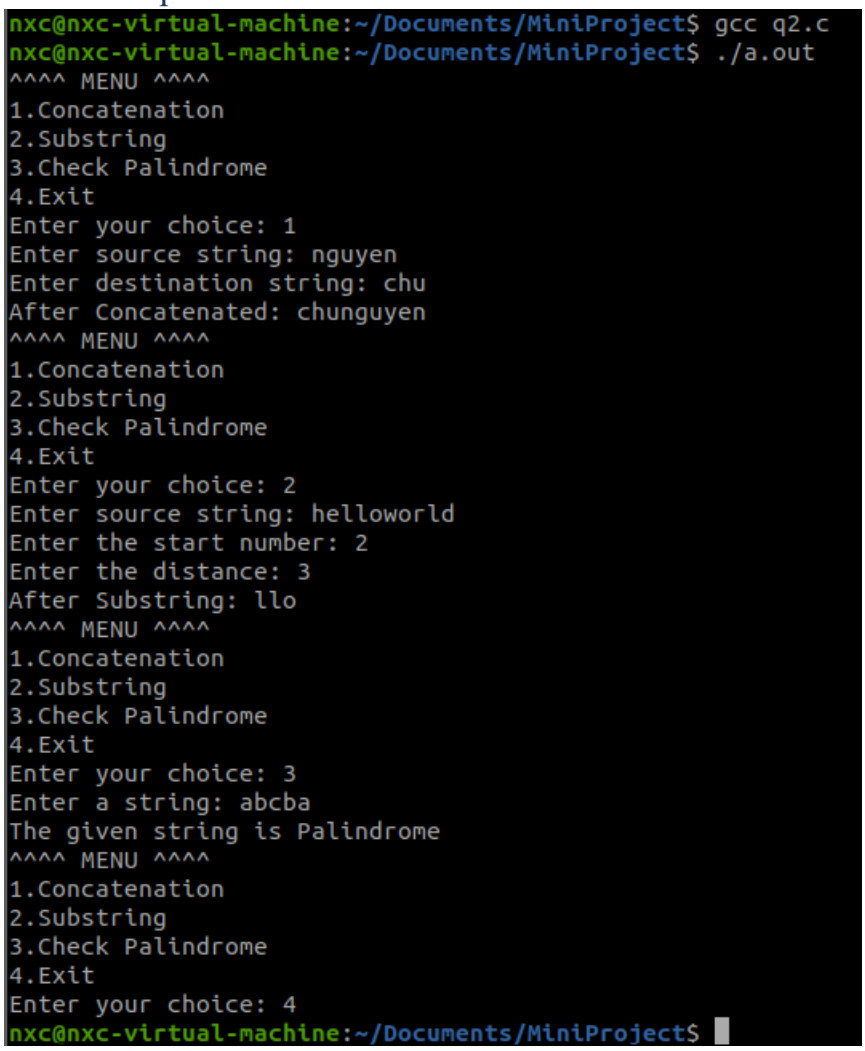
```
{
    int len_pd = len(pd);
    int i;
    for (i=0; i<len(ps); i++)
    {
        *(pd+i+len_pd) = *(ps+i);
    }
    *(pd+len_pd+i) = '\0';
}
//string =          nguyen123
//begin = 3 = y
//distance = 3

//ptr = 0 = n
//=>ptr += str_begin => ptr = 3
//result =      yen
void substring(char *pd, char *ps, int str_begin, int distance)
{
    ps += str_begin;
    int i;
    for (i=0; i<distance; i++)
    {
        *(pd+i) = *(ps+i);
    }
    *(pd+i) = '\0';
}
//012345
//abcba\0
//abccba
int is_palindrome(char *ptr)
```

```
{
    int i = 0;
    int n = len(ptr);
    for (int i=0; i<n/2; i++)
    {
        if (*(ptr+i) == *(ptr+n-1-i))
        {
            i++;
        }
        else
        {
            return 0;
        }
    }
    return 1;
}
int main()
{
    char str_s[30] = "";
    char str_d[30] = "";
    int start_number;
    int distance;
    int choice;
    while(1)
    {
        printf("^^^^ MENU ^^^^\n");
        printf("1.Concatenation\n");
        printf("2.Substring\n");
        printf("3.Check Palindrome\n");
        printf("4.Exit\n");
```

```
printf("Enter your choice: ");
scanf("%d", &choice);
switch(choice)
{
    case 1:
        printf("Enter source string: ");
        scanf("%s", str_s);
        printf("Enter destination string: ");
        scanf("%s", str_d);
        concatenate(str_d, str_s);
        printf("After Concatenated: ");
        printf("%s\n",str_d);
        break;
    case 2:
        printf("Enter source string: ");
        scanf("%s", str_s);
        printf("Enter the start number: ");
        scanf("%d", &start_number);
        printf("Enter the distance: ");
        scanf("%d", &distance);
        substring(str_d, str_s, start_number, distance);
        printf("After Substring: ");
        printf("%s\n",str_d);
        break;
    case 3:
        printf("Enter a string: ");
        scanf("%s", str_s);
        if (is_palindrome(str_s))
        {
            printf("The given string is Palindrome\n");
```

```
            }
            else
            {
                printf("The given string is Palindrome\n");
            }
            break;
        case 4:
            return 0;
            break;
        default:
            printf("Please enter a valid number\n");
            break;
    }
  }
}
```

Result of question 2:



Solution of question 3:

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>


typedef struct

{

    char name[30];

    int score;

} Student;

Student *add_student(Student *old_ptr, int *new_size)

{
```

```
    ++*new_size;
    Student *ptr_new = (Student *)realloc(old_ptr, *new_size*sizeof(Student));
    return ptr_new;
}
void update_student(Student *ptr, int size)
{
    char ch[30];
    printf("Enter student's name: ");
    scanf("%s", ch);
    for (int i=0; i<size; i++)
    {
        if (strcmp((ptr+i)->name, ch)==0)
        {
            printf("Enter new student's score: ");
            scanf("%d", &(ptr+i)->score);
            return;
        }
    }
    printf("Could not find out that student\n");
}
Student *delete_student(Student *ptr, int *size)
{
    char ch[30];
    printf("Enter student's name: ");
    scanf("%s", ch);
    for (int i=0; i<*size; i++)
    {
        if (strcmp((ptr+i)->name, ch)==0)
        {
            printf("Found out\n");
```

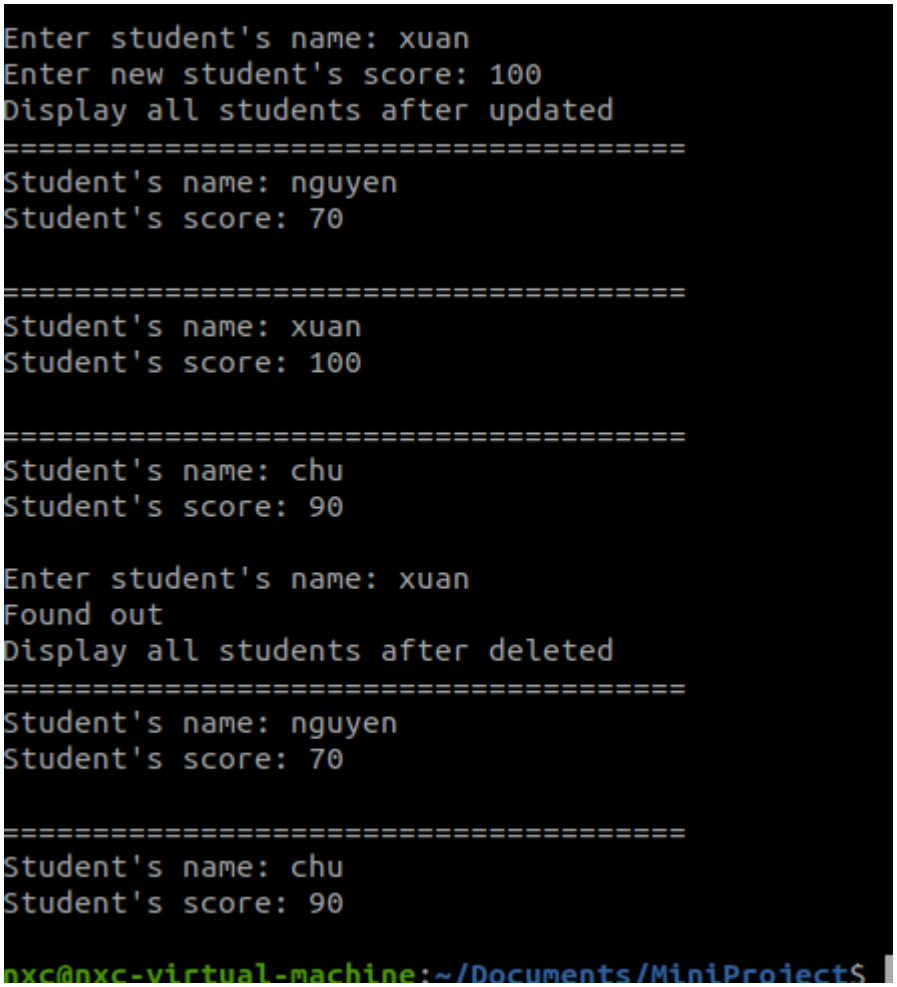
```
            for (int j=i; j<*size; j++)
            {
                *(ptr+j) = *(ptr+j+1);
            }
            --*size;
            Student *ptr_new = (Student *)realloc(ptr, *size*sizeof(Student));
            return ptr_new;
        }
    }
    printf("Could not find out that student\n");
}
int main()
{
    int n;
    printf("Enter number of students: ");
    scanf("%d",&n);

    Student *ptr_s = (Student*) malloc (n*sizeof(Student));
    //import Student
    for (int i=0; i<n; i++)
    {
        printf("Enter student's name: ");
        scanf("%s", (ptr_s+i)->name);
        printf("Enter student's score: ");
        scanf("%d", &(ptr_s+i)->score);
    }
    //display
    printf("Display all students\n");
    for (int i=0; i<n; i++)
    {
```

```
    printf("=====================================\n");
    printf("Student's name: %s\n", (ptr_s+i)->name);
    printf("Student's score: %d\n", (ptr_s+i)->score);
    printf("\n");
}
//add new Student
Student *new_ptr_s;
ptr_s = add_student(ptr_s, &n);
printf("Enter new student's name: ");
scanf("%s", (ptr_s+n-1)->name);
printf("Enter new student's score: ");
scanf("%d", &(ptr_s+n-1)->score);
//display all student after added
printf("Display all students after added\n");
for (int i=0; i<n; i++)
{
    printf("=====================================\n");
    printf("Student's name: %s\n", (ptr_s+i)->name);
    printf("Student's score: %d\n", (ptr_s+i)->score);
    printf("\n");
}
//update student
update_student(ptr_s, n);
//display all student after updated
printf("Display all students after updated\n");
for (int i=0; i<n; i++)
{
    printf("=====================================\n");
    printf("Student's name: %s\n", (ptr_s+i)->name);
    printf("Student's score: %d\n", (ptr_s+i)->score);
```

```
        printf("\n");
    }
    //delete student
    ptr_s = delete_student(ptr_s, &n);
    //display all student after deleted
    printf("Display all students after deleted\n");
    for (int i=0; i<n; i++)
    {
        printf("=====================================\n");
        printf("Student's name: %s\n", (ptr_s+i)->name);
        printf("Student's score: %d\n", (ptr_s+i)->score);
        printf("\n");
    }
    free(ptr_s);
    return 0;
}
```

Result of question 3:

```
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter number of students: 2
Enter student's name: nguyen
Enter student's score: 70
Enter student's name: xuan
Enter student's score: 80
Display all students
====================================
Student's name: nguyen
Student's score: 70

====================================
Student's name: xuan
Student's score: 80

Enter new student's name: chu
Enter new student's score: 90
Display all students after added
====================================
Student's name: nguyen
Student's score: 70

====================================
Student's name: xuan
Student's score: 80

====================================
Student's name: chu
Student's score: 90

Enter student's name: xuan
Enter new student's score: 100
Display all students after updated
====================================
Student's name: nguyen
Student's score: 70
```



Solution of question 4:

```
#include <stdio.h>


float convert_C_to_F(float c)
{
    return ((c * 1.8) +32);
}
float convert_C_to_K(float c)
{
    return (c + 273.15);
}


float convert_F_to_C(float f)
{
```

```
    return ((f - 32)*1.8);
}
float convert_F_to_K(float f)
{
    return ((f - 32)/1.8+273.15);
}

float convert_K_to_C(float k)
{
    return (k - 273.15);
}
float convert_K_to_F(float k)
{
    return ((k*9.5) - 459.67);
}
int main()
{
    float n;
    FILE *fp;
    //open file mode append
    fp = fopen("./data.txt", "a");
    if (fp < 0) {
        printf("Error in opening file.\n");
        return 0;
    }
    printf("Enter degree Celsius value: ");
    scanf("%f", &n);
    float result = convert_C_to_F(n);
    if(result < -459.67)
    {
```

```
        printf("Enter the temp.\n");
    }
    else
    {
        fprintf(fp,"C to F is %.2f\n",result);

    }
    result = convert_C_to_K(n);
    //0 kelvin or -273.15 C
    if(result < 0)
    {
        printf("Enter the temp.\n");
    }
    else
    {
        fprintf(fp,"C to K is %.2f\n",result);

    }
    // close file
    fclose(fp);
    printf("Data saved in file.\n");

    return 0;
}
```

Result of question 4:

```
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter degree Celsius value: 100
Data saved in file.
nxc@nxc-virtual-machine:~/Documents/MiniProject$ cat ./data.txt
C to F is 212.00
C to K is 373.15
nxc@nxc-virtual-machine:~/Documents/MiniProject$
```

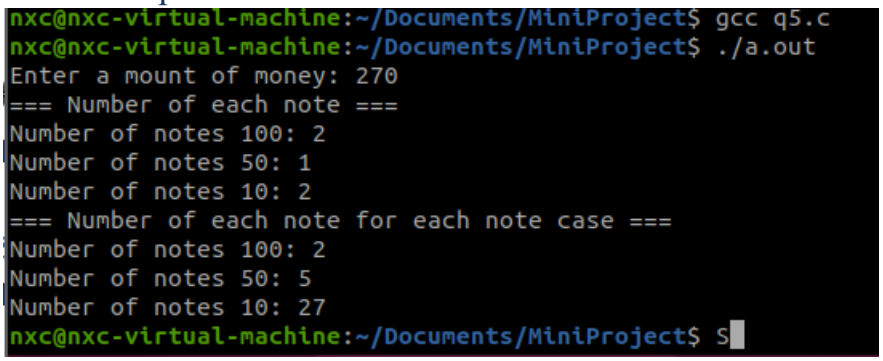
Solution of question 5:

```
#include<stdio.h>
```

```
int main()
{
    int n;
    do
    {
        printf("Enter a mount of money: ");
        scanf("%d", &n);
    } while (n%10);

    int n100;
    int n50;
    int n10;

    //n = 270
    //  n/100 -> n = 2
    //  n%100 -> m = 70
    //  m/50  -> m = 1
    //  m%50  -> k = 20
    //  k/10 -> k = 2
    printf("=== Number of each note ===\n");
    n100 = n/100;
    printf("Number of notes 100: %d\n", n100);
    n50 = (n%100)/50;
    printf("Number of notes 50: %d\n", n50);
    n10 = (n%100)%50/10;
    printf("Number of notes 10: %d\n", n10);

    printf("=== Number of each note for each note case ===\n");
    printf("Number of notes 100: %d\n", n/100);
```

```
    printf("Number of notes 50: %d\n", n/50);
    printf("Number of notes 10: %d\n", n/10);


    return 0;
}
```

Result of question 5:



Solution of question 6:

```
#include<stdio.h>


int main()
{
    int n;
    float fine = 0;
    printf("Enter your day late: ");
    scanf("%d", &n);
    if (n>30)
    {
        printf("Your membership will be canceled\n");
        return 0;
    }
    else
    {
        switch(n)
        {
            case 1 ... 5:
```

```
            fine += (n*0.5);
            printf("%f\n", fine);
            break;
        case 6 ... 10:
            fine = 0.5*5 + 1*(n-5);
            printf("%f\n", fine);
            break;
        case 11 ... 30:
            fine = 0.5*5 + 1*5 + 5*(n-10);
            printf("%f\n", fine);
            break;
        default:
            printf("Enter a valid positive number\n");
            break;
    }
    return 0;
  }
}
```

Result of question 6:

```
nxc@nxc-virtual-machine:~/Documents/MiniProject$ gcc q6.c
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter your day late: 4
2.000000
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter your day late: 6
3.500000
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter your day late: 20
57.500000
nxc@nxc-virtual-machine:~/Documents/MiniProject$ 31
31: command not found
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter your day late: 31
Your membership will be canceled
nxc@nxc-virtual-machine:~/Documents/MiniProject$
```

Mini Project C++

Solution of question 1:

```
#include <iostream>

#include <vector>
```

```
using namespace std;

class Student
{
private:
    int ID;
    string name;
    int age;
    int marks[3];
    float avg = 0;

public:
    void get_info()
    {
        cout << "Enter name of student: ";
        cin >> name;
        cout << "Enter ID of student: ";
        cin >> ID;
        cout << "Enter age of student: ";
        cin >> age;
        for (int i = 0; i < 3; i++)
        {
            cout << "Enter mark of subject " << i + 1 << ": ";
            cin >> marks[i];
        }
        avg = average();
    }

    float average()
    {
```

```
        return (marks[0] + marks[1] + marks[2]) / 3.0;
    }

    void display()
    {
        cout << "Name: " << name << endl;
        cout << "ID: " << ID << endl;
        cout << "Age: " << age << endl;
        for (int i = 0; i < 3; i++)
        {
            cout << "Marks of subject " << i + 1 << ": " << marks[i] << endl;
        }
        cout << "Average is: " << average() << endl;
    }
    int get_id()
    {
        return ID;
    }
    int get_average()
    {
        return avg;
    }
    void change_mark(int new_mark, int index)
    {
        marks[index] = new_mark;
    }
};

int main()
{
```

```
int n;
int S_ID;
int new_mark, pos;
int index = 0;
int choice, size;
float max_avg;
//////
vector<Student> vector_student;

while (1)
{
    cout << "1.Add student." << endl;
    cout << "2.Display all student." << endl;
    cout << "3.Display Average all student." << endl;
    cout << "4.Display Topper" << endl;
    cout << "5.Search student by ID." << endl;
    cout << "6.Update student marks." << endl;
    cout << "7.Exit." << endl;
    cout << "Enter your choice, please:";
    try
    {
        cin >> choice;
        if (cin.fail())
        {
            throw 0;
        }
        switch (choice)
        {
        case 1:
            cout << "Enter number of student: ";
```

```
    cin >> n;

    for (int i = 0; i < n; i++)
    {
        Student new_student;
        new_student.get_info();
        vector_student.push_back(new_student);
    }
    break;
case 2:
    size = vector_student.size();
    for (int i = 0; i < size; i++)
    {
        vector_student[i].display();
    }
    break;

case 3:
    cout << "key 3" << endl;
    size = vector_student.size();
    for (int i = 0; i < size; i++)
    {
        cout << "Average marks of student " << i + 1 << endl;
        cout << vector_student[i].get_average() << endl;
    }
    break;
case 4:
    max_avg = vector_student[0].get_average();
    size = vector_student.size();
    for (int i = 1; i < size; i++)
```

```
    {
        if (vector_student[i].get_average() > max_avg)
        {
            max_avg = vector_student[i].get_average();
            index = i;
        }
    }

    cout << "Student with the highest average:" << endl;
    vector_student[index].display();
    cout << "Max Average: " << max_avg << endl;
    break;
case 5:
    size = vector_student.size();
    cout << "Enter ID of student: ";
    cin >> S_ID;
    for (int i = 0; i < size; i++)
    {

        if (S_ID == vector_student[i].get_id())
        {
            cout << "Student is: " << endl;
            vector_student[i].display();
        }
    }
    break;
case 6:
    size = vector_student.size();
    cout << "Enter ID of student: ";
    cin >> S_ID;
```

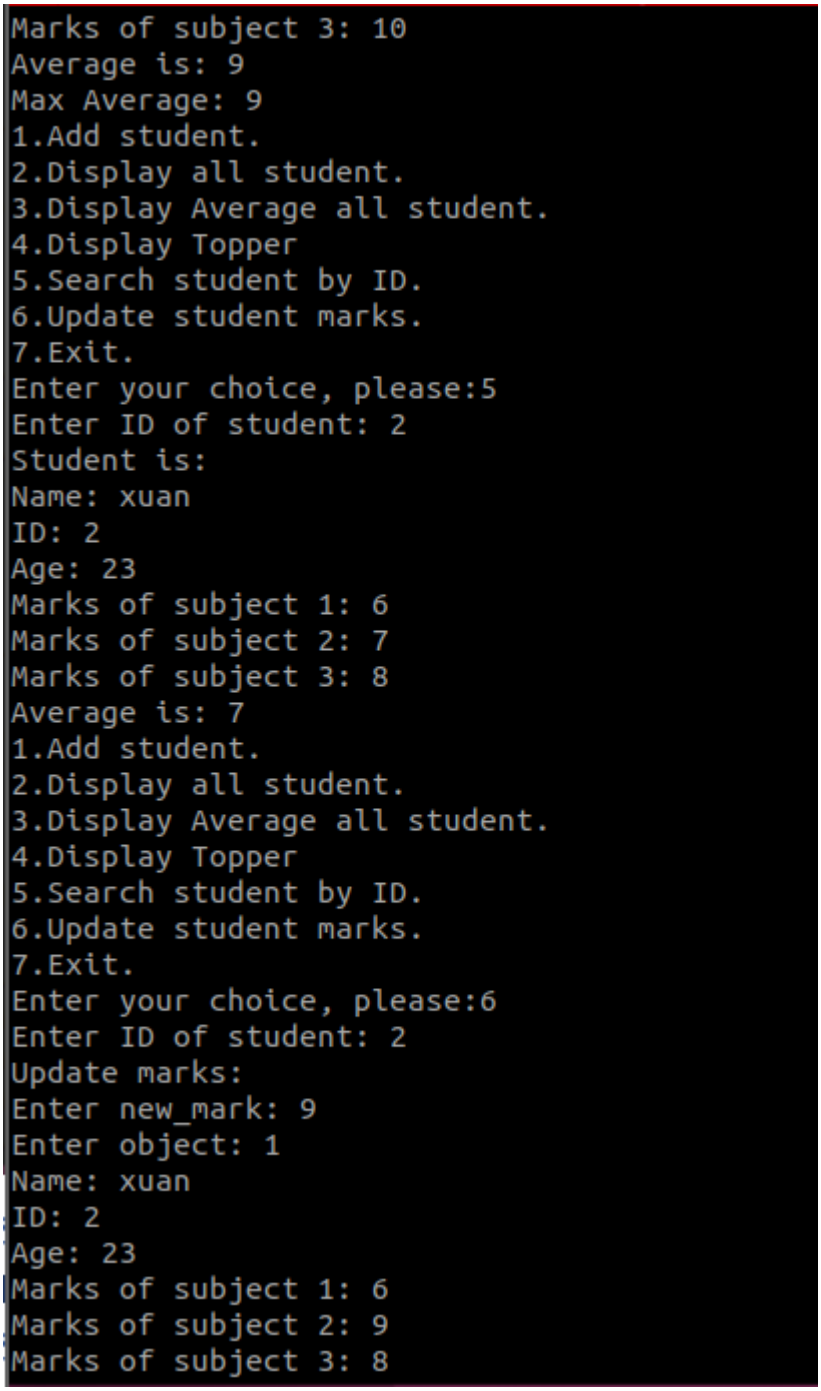
```
            for (int i = 0; i < size; i++)
            {

                if (S_ID == vector_student[i].get_id())
                {
                    cout << "Update marks: " << endl;
                    cout << "Enter new_mark: ";
                    cin >> new_mark;
                    cout << "Enter object: ";
                    cin >> pos;
                    vector_student[i].change_mark(new_mark, pos);
                    vector_student[i].display();
                }
            }
            break;
        case 7:
            return 0;
            break;
        default:
            cout << "Enter your choice, please!!!" << endl;
            break;
        }
    }
    catch (int e)
    {
        cin.clear();
        cin.ignore(100, '\n');
        cout << "Please enter valid number!!!!" << endl;
    }
}
```

```
}
```

Result of question 1:

```
nxc@nxc-virtual-machine:~/Documents/MiniProject$ g++ q1.cpp
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
1.Add student.
2.Display all student.
3.Display Average all student.
4.Display Topper
5.Search student by ID.
6.Update student marks.
7.Exit.
Enter your choice, please:1
Enter number of student: 2
Enter name of student: nguyen
Enter ID of student: 1
Enter age of student: 23
Enter mark of subject 1: 8
Enter mark of subject 2: 9
Enter mark of subject 3: 10
Enter name of student: xuan
Enter ID of student: 2
Enter age of student: 23
Enter mark of subject 1: 6
Enter mark of subject 2: 7
Enter mark of subject 3: 8
1.Add student.
2.Display all student.
3.Display Average all student.
4.Display Topper
5.Search student by ID.
6.Update student marks.
7.Exit.
Enter your choice, please:2
Name: nguyen
ID: 1
Age: 23
Marks of subject 1: 8
Marks of subject 2: 9
Marks of subject 3: 10
Average is: 9
```

```
Marks of subject 3: 10
Average is: 9
Name: xuan
ID: 2
Age: 23
Marks of subject 1: 6
Marks of subject 2: 7
Marks of subject 3: 8
Average is: 7
1.Add student.
2.Display all student.
3.Display Average all student.
4.Display Topper
5.Search student by ID.
6.Update student marks.
7.Exit.
Enter your choice, please:3
key 3
Average marks of student 1
9
Average marks of student 2
7
1.Add student.
2.Display all student.
3.Display Average all student.
4.Display Topper
5.Search student by ID.
6.Update student marks.
7.Exit.
Enter your choice, please:4
Student with the highest average:
Name: nguyen
ID: 1
Age: 23
Marks of subject 1: 8
Marks of subject 2: 9
Marks of subject 3: 10
Average is: 9
```



Solution of question 2:

```
#include <iostream>

#include <fstream>

#include <ctime>

using namespace std;


class BankAccount

{
```

```
private:
int acc_no;
string name;
string acc_type;
int balance;
int password;
public:
//constructor
BankAccount(int n=0, string un="", string at="", int pw=0, int b=0)
    :acc_no(n),name(un),acc_type(at),balance(b),password(pw){}
int check_password(int pass)
{
    if (pass == password)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
int check_account_number(int num)
{
    if (num == acc_no)
    {
        return 1;
    }
    else
    {
        return 0;
```

```
    }
}
void deposit(int value)
{
    if (value>0)
    {
        cout<<"### Deposit Successfully ###\n"<<endl;
        balance+=value;
        transaction_record("deposit", value);
    }
    else
    {
        cout<<"!!! Error: Enter a valid number\n"<<endl;
    }
}
void withdraw(int value)
{
    if (value>0 && value<=balance)
    {
        cout<<"### Withdraw Successfully ###\n"<<endl;
        balance-=value;
        transaction_record("withdraw", value);
    }
    else if (value<0)
    {
        cout<<"!!! Error: Enter a valid number\n"<<endl;
    }
    else if (value>balance)
    {
        cout<<"!!! Error: Not enough to withdraw\n"<<endl;
```

```
    }
}
void transaction_record(string transac_type, int value)
{
    //current time
    time_t now = time(0);
    //open file
    fstream logfile;
    logfile.open("./log.txt", ios::out|ios::app);
    if (logfile.is_open())
    {
        logfile<<to_string(now)+" "+transac_type+" "+to_string(value)+"\n";
        logfile.close();
    }
}
void transac_history()
{
    fstream logfile;
    logfile.open("./log.txt", ios::in);
    if (logfile.is_open())
    {
        string line;
        cout<<"=== Diplay Log File ==="<<endl;
      while(!logfile.eof())
      {
                getline(logfile,line);
                //display in terminal
                cout<<line<<endl;
      }
      logfile.close();
```

```
        }
        else
        {
            cout<<"!!! Error: Cannot open log file"<<endl;
        }
    }
    void display()
    {
        {
            cout<<"=== Bank Account Detail ==="<<endl;
            cout<<"Account Number: "<<acc_no<<endl;
            cout<<"Name: "<<name<<endl;
            cout<<"Account Type: "<<acc_type<<endl;
            cout<<"Balance: "<<balance<<endl;
            cout<<endl;
        }
    }
};

int main()
{
    BankAccount ba(1234, "nguyen", "saving", 1234);
    int num;
    int pass;
    int choice;
    int amount;
    //login
    do
    {
        cout<<"Enter your account number: ";
```

```
    cin>>num;
    cout<<"Enter your password: ";
    cin>>pass;
} while (!ba.check_password(pass) || !ba.check_account_number(num));

//menu
while(1)
{
    cout<<"*** *** MENU *** ***"<<endl;
    cout<<"1.Display Account Detail"<<endl;
    cout<<"2.Deposit"<<endl;
    cout<<"3.Withdraw"<<endl;
    cout<<"4.Display Transaction History"<<endl;
    cout<<"5.Exit"<<endl;
    cout<<"Choice: ";
    cin>>choice;
    switch(choice)
    {
        case 1:
            ba.display();
            break;
        case 2:
            cout<<"Enter an amount: ";
            cin>>amount;
            ba.deposit(amount);
            break;
        case 3:
            cout<<"Enter an amount: ";
            cin>>amount;
            ba.withdraw(amount);
```

```
            break;
        case 4:
            ba.transac_history();
            break;
        case 5:
            return 0;
            break;
        default:
            cout<<"Pleas enter a valid value"<<endl;
            break;
      }
   }
}
```

Result of question 2:

```
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
Enter your account number: 123
Enter your password: 1234
Enter your account number: 1234
Enter your password: 1234
*** *** MENU *** ***
1.Display Account Detail
2.Deposit
3.Withdraw
4.Display Transaction History
5.Exit
Choice: 1
=== Bank Account Detail ===
Account Number: 1234
Name: nguyen
Account Type: saving
Balance: 0

*** *** MENU *** ***
1.Display Account Detail
2.Deposit
3.Withdraw
4.Display Transaction History
5.Exit
Choice: 2
Enter an amount: 200
### Deposit Successfully ###

*** *** MENU *** ***
1.Display Account Detail
2.Deposit
3.Withdraw
4.Display Transaction History
5.Exit
Choice: 3
Enter an amount: 40
### Withdraw Successfully ###
```

```
### Withdraw Successfully ###

*** *** MENU *** ***
1.Display Account Detail
2.Deposit
3.Withdraw
4.Display Transaction History
5.Exit
Choice: 4
=== Diplay Log File ===
1720354946 deposit 200
1720354951 withdraw 40

*** *** MENU *** ***
1.Display Account Detail
2.Deposit
3.Withdraw
4.Display Transaction History
5.Exit
Choice: 5
nxc@nxc-virtual-machine:~/Documents/MiniProject$ cat ./log.txt
1720354946 deposit 200
1720354951 withdraw 40
nxc@nxc-virtual-machine:~/Documents/MiniProject$
```

Solution of question 3:

```
#include<iostream>

#include<cstring>

#include<vector>

#include<algorithm>

using namespace std;


class Book

{

    private:

    int id;

    string title;

    string author;

    string genre;

    int quantity_available;

    float price_copy;
```

```
public:
//constructor(
Book(int uid=0,string utitle="",string uauthor="",
        string ugenre="", int uqa=0, float ppc=0) :
    id(uid),
    title(utitle),
    author(uauthor),
    genre(ugenre),
    quantity_available(uqa),
    price_copy(ppc)
    {}
void get_info()
      {
              cout << "===== Enter Book Information =====<<endl;
              cout << "Enter Book ID : ";
              cin >> id;
              cout << "Enter Title: ";
              cin >> title;
              cout << "Enter author: ";
              cin >> author;
              cout << "Enter genre: ";
              cin >> genre;
              cout << "Enter Quantity Available: ";
              cin >> quantity_available;
              cout << "Enter Price per Copy: ";
              cin >> price_copy;
      }
      string get_title()
      {
          return  title;
```

```
        }
        void update_quantity(int value)
        {
            quantity_available += value;
        }
    float calculate_revenue()
    {
       return (float)quantity_available*price_copy;
    }
        void display()
        {
                cout << "===== Book Information =====<<endl;
                cout << "Book ID : " << id << endl;
                cout << "Enter Title: " << title << endl;
                cout << "Enter author: " << author << endl;
                cout << "Enter genre: " << genre << endl;
                cout << "Enter Quantity Available: " << quantity_available << endl;
                cout << "Enter Price per Copy: " << price_copy << endl;
                cout<<endl;
        }
};

int main()
{
    vector<Book> vect;
    Book b(0);

    int choice;
    int len;
    int pos = 0;
```

```
float max = 0;
int flag = 0;
string title_book;
int new_quantity;
while(1)
{
    cout<<"^^^^^^^^^^ MENU ^^^^^^^^^^"<<endl;
    cout<<"1.Add Book"<<endl;
    cout<<"2.Display All Books"<<endl;
    cout<<"3.Display Revenue"<<endl;
    cout<<"4.Display Bestselling Book"<<endl;
    cout<<"5.Search Book By Title"<<endl;
    cout<<"6.Update Book Quantity"<<endl;
    cout<<"7.Exit"<<endl;
    cout<<"Enter your choice: ";
    cin>>choice;

    switch(choice)
    {
        case 1:
            b.get_info();
            vect.push_back(b);
            break;
        case 2:
            len = vect.size();
            for (int i=0; i<len; i++)
            {
                vect[i].display();
            }
            break;
```

```
case 3:
    len = vect.size();
    for (int i=0; i<len; i++)
    {
        cout<<"Revenue of "<<vect[i].get_title()<<" : ";
        cout<<vect[i].calculate_revenue()<<endl;
    }
    cout<<endl;
    break;
case 4:
    cout<<"Best Seller: "<<endl;
    len = vect.size();
    max = vect[0].calculate_revenue();
         for (int i=0; i<len; i++)
         {
        if (vect[i].calculate_revenue() > max)
        {
            max = vect[i].calculate_revenue();
            pos = i;
        }
         }
    vect[pos].display();
    break;
case 5:
    cout<<"Enter the book's title: ";
    cin>>title_book;
         for (int i=0; i<len; i++)
         {
        if (title_book.compare(vect[i].get_title())==0)
        {
```

```
            flag = 1;
                    vect[i].display();
            break;
        }
         }
         if (!flag)
         {
          cout<<"Counld Not Find Out That Book's Title"<<endl;
         }
         flag = 0;
    break;
case 6:
    cout<<"Enter the book's title: ";
    cin>>title_book;
         for (int i=0; i<len; i++)
         {
        if (title_book.compare(vect[i].get_title())==0)
        {
                flag = 1;
                  cout<<"Enter new quantity: ";
                  cin>>new_quantity;
                vect[i].update_quantity(new_quantity);
                vect[i].display();
            break;
        }
         }
         if (!flag)
         {
          cout<<"Counld Not Find Out That Book's Title"<<endl;
         }
```

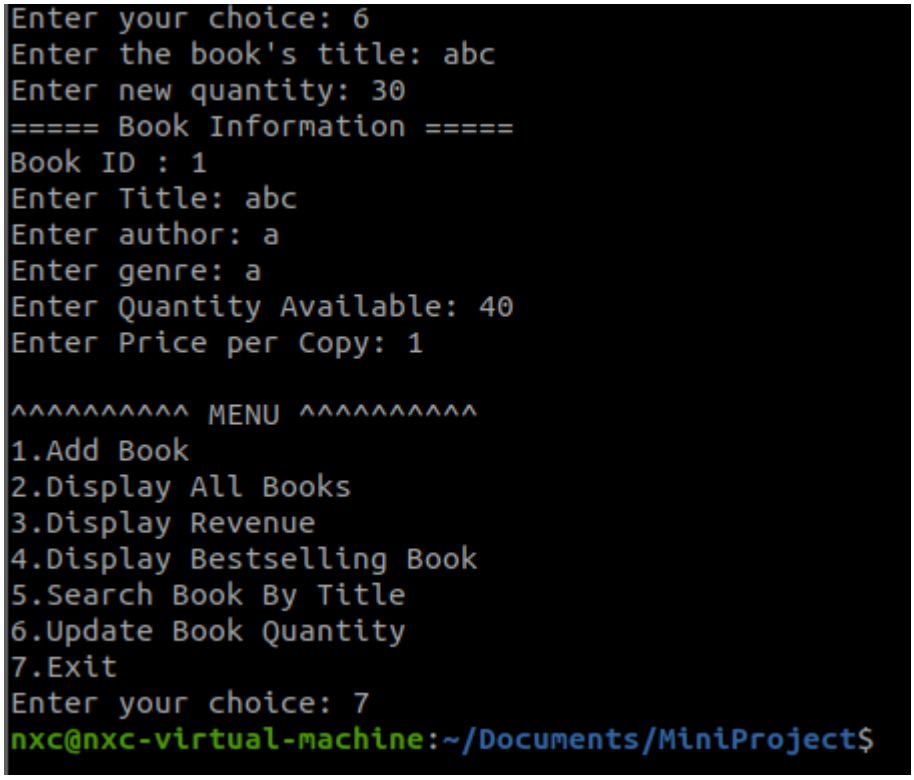
```
                flag = 0;
            break;
        case 7:
            return 0;
            break;
        default:
            cout<<"Please Enter A Valid Value"<<endl;
            break;
    }
  }
}
```

Result of question 3:

```
nxc@nxc-virtual-machine:~/Documents/MiniProject$ g++ q3.cpp
nxc@nxc-virtual-machine:~/Documents/MiniProject$ ./a.out
^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 1
===== Enter Book Information =====
Enter Book ID : 1
Enter Title: abc
Enter author: a
Enter genre: a
Enter Quantity Available: 10
Enter Price per Copy: 1
^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 1
===== Enter Book Information =====
Enter Book ID : 2
Enter Title: asd
Enter author: b
Enter genre: b
Enter Quantity Available: 10
Enter Price per Copy: 2
^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
```

```
7.Exit
Enter your choice: 2
===== Book Information =====
Book ID : 1
Enter Title: abc
Enter author: a
Enter genre: a
Enter Quantity Available: 10
Enter Price per Copy: 1

===== Book Information =====
Book ID : 2
Enter Title: asd
Enter author: b
Enter genre: b
Enter Quantity Available: 10
Enter Price per Copy: 2

^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 3
Revenue of abc : 10
Revenue of asd : 20

^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 4
Best Seller:
===== Book Information =====
```

```
Enter your choice: 4
Best Seller:
===== Book Information =====
Book ID : 2
Enter Title: asd
Enter author: b
Enter genre: b
Enter Quantity Available: 10
Enter Price per Copy: 2

^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 5
Enter the book's title: asd
===== Book Information =====
Book ID : 2
Enter Title: asd
Enter author: b
Enter genre: b
Enter Quantity Available: 10
Enter Price per Copy: 2

^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 6
Enter the book's title: abc
Enter new quantity: 30
===== Book Information =====
Book ID : 1
```


Enter your choice: 6
Enter the book's title: abc
Enter new quantity: 30
===== Book Information =====
Book ID : 1
Enter Title: abc
Enter author: a
Enter genre: a
Enter Quantity Available: 40
Enter Price per Copy: 1
^^^^^^^^^^ MENU ^^^^^^^^^^
1.Add Book
2.Display All Books
3.Display Revenue
4.Display Bestselling Book
5.Search Book By Title
6.Update Book Quantity
7.Exit
Enter your choice: 7
nxc@nxc-virtual-machine:~/Documents/MiniProject$