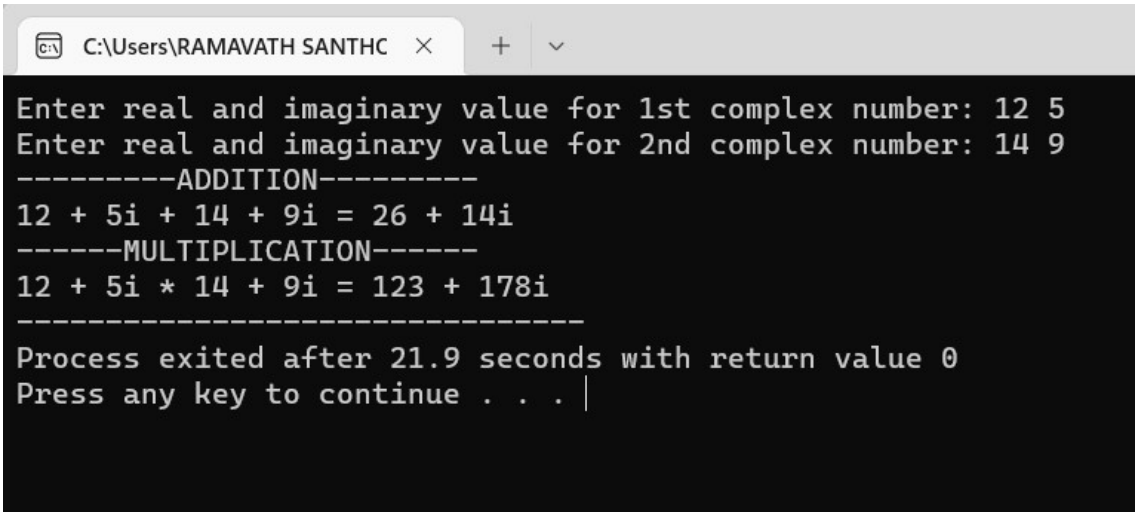
**********ASSIGNMENT 6**********

NAME: RAMAVATH SANTHOSH ROLL NO: 22MCF1R40 MCA

1. Create a structure for complex number, do addition and multiplication of two complex number

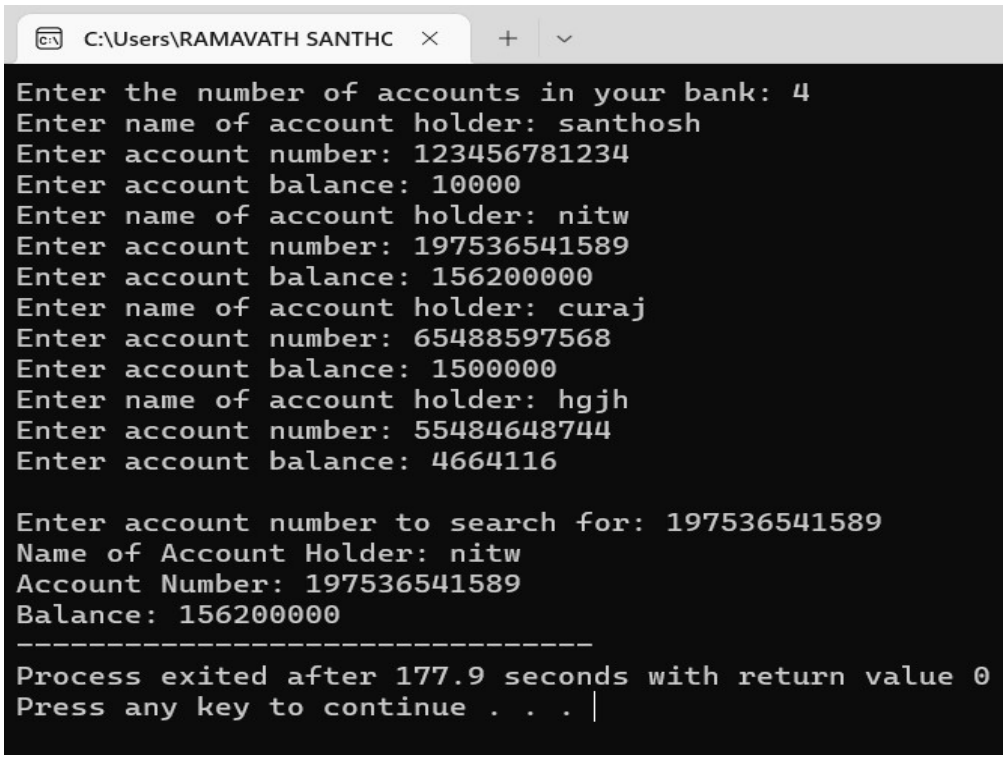
```
//Q1  Create a structure for complex number, do addition and multiplication of two complex number
#include <iostream>

using namespace std;

struct complex
{
	int real, img;
};

void printComplex(complex c)
{
	cout << c.real << " ";
	if(c.img < 0) cout << "- " << -c.img;
	else cout << "+ " << c.img;
	cout << "i";
}

int main()
{
	complex c1, c2;
	cout << "Enter real and imaginary value for 1st complex number: "; cin >> c1.real >> c1.img;
	cout << "Enter real and imaginary value for 2nd complex number: "; cin >> c2.real >> c2.img;

	cout << "---------ADDITION---------\n";
	printComplex(c1); cout << " + "; printComplex(c2);
	complex c3;
	c3.real = c1.real + c2.real;
	c3.img = c1.img + c2.img;
	cout << " = "; printComplex(c3);

	cout << "\n------MULTIPLICATION------\n";
	printComplex(c1); cout << " * "; printComplex(c2);
	c3.real = c1.real * c2.real - c1.img * c2.img;
	c3.img = c1.img * c2.real + c1.real * c2.img;
	cout << " = "; printComplex(c3);
	return 0;
}
```




2. Create a structure for Bank account with name, account number and balance. Create a list of account holders using array of structures and get input from user to populate them. Display the account details given the account number.

```
//Q2.    Create a structure for Bank account with name, account number and balance.
//Create a list of account holders using array of structures and get input from user to
populate them.
// Display the account details given the account number.

#include <iostream>
#include <cstring>

using namespace std;

struct account
{
        string name;
        string accno;
        int balance;
};

int main()
{
        account accs[10];
        int n;
        cout << "Enter the number of accounts in your bank: "; cin >> n;
        cin.get();
        for(int i = 0; i < n; i++)
        {
                cout << "Enter name of account holder: "; getline(cin, accs[i].name);
                cout << "Enter account number: "; cin >> accs[i].accno;
```

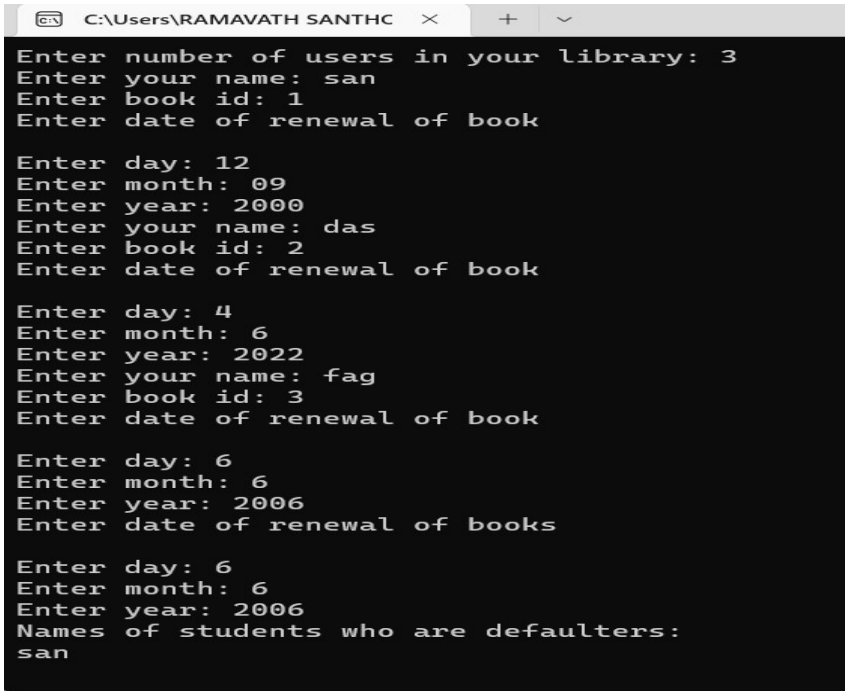
```
            cout << "Enter account balance: "; cin >> accs[i].balance;
            cin.get();
        }

        string accno;
        cout << "\nEnter account number to search for: "; cin >> accno;
        bool found = false;
        for(int i = 0; i < n; i++)
        {
            if(accs[i].accno == accno)
            {
                cout << "Name of Account Holder: " << accs[i].name << endl;
                cout << "Account Number: " << accs[i].accno << endl;
                cout << "Balance: " << accs[i].balance;
                found = true;
                break;
            }
        }

        if(!found)
            cout << "Account not found";
        return 0;
}
```



3. Create a structure for library users with name, book id, date of renewal. Create a list of library users and update their entries from user. Find out on a particular day who are the defaulters. Assume only one book is issued per person.

```
//Q3 3. Create a structure for library users with name, book id, date of renewal.
//Create a list of library users and update their entries from user.
//Find out on a particular day who are the defaulters. Assume only one book is issued per person.

#include <iostream>
#include <string>

using namespace std;

struct date{
	int day;
	int month;
	int year;
};

struct library{
	string name;
	string book_id;
	date day_of_renewal;
};

void getDate(date &d)
{
	cout << endl;
	cout << "Enter day: "; cin >> d.day;
	cout << "Enter month: "; cin >> d.month;
	cout << "Enter year: "; cin >> d.year;
}

int compareDate(date d1, date d2)
{
	if(d1.year == d2.year)
		if(d1.month == d2.month)
			if(d1.day == d2.day) return 0;
			else if(d1.day < d2.day) return -1;
			else return 1;
		else if(d1.month < d2.month) return -1;
		else return 1;
	else if(d1.year < d2.year) return -1;
	return 1;
}

int main()
```

```
{
	int n;

	cout << "Enter number of users in your library: "; cin >> n;
	library users[n];

	for(int i = 0; i < n; i++)
	{
		getchar();
		cout << "Enter your name: "; getline(cin, users[i].name);
		cout << "Enter book id: "; cin >> users[i].book_id;
		getchar();
		cout << "Enter date of renewal of book\n";
getDate(users[i].day_of_renewal);
	}

	date returnDate, dummy;

	cout << "Enter date of renewal of books\n"; getDate(returnDate);

	cout << "Names of students who are defaulters:\n";
	for(int i = 0; i < n; i++)
		if(compareDate(returnDate, users[i].day_of_renewal) > 0)
			cout << users[i].name << endl;
	return 0;
}
```

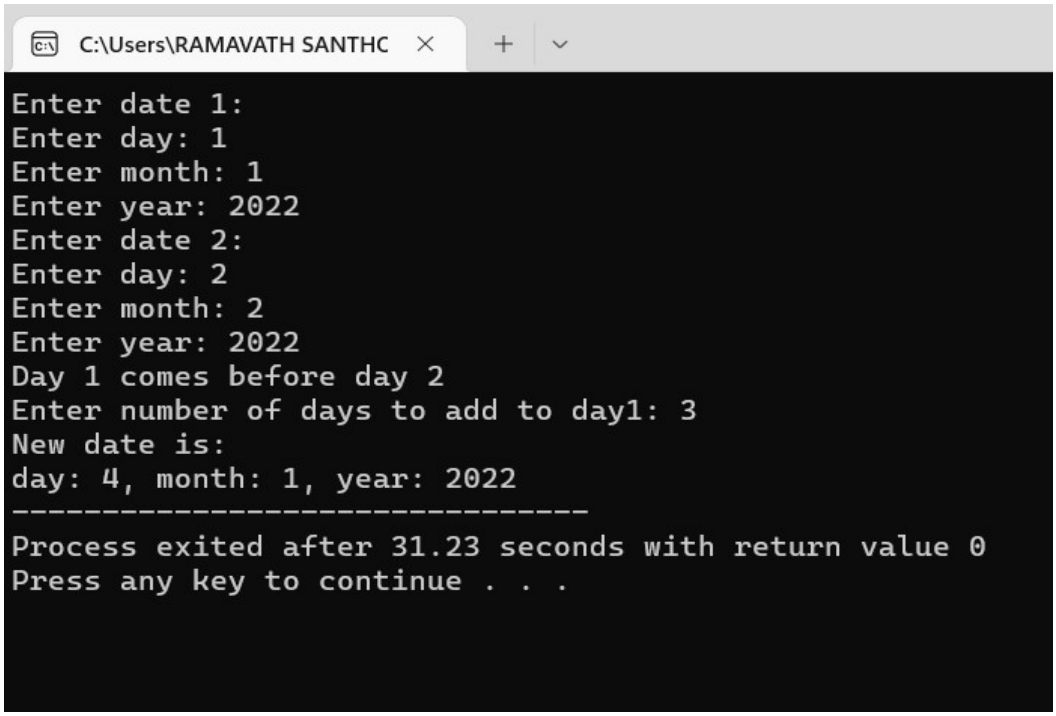


4. Create a structure for calender date with day, month and year. Find if given two days are equal, or which is earlier. Write a function to add days to the date structure to form the new date. Assume no leap year.

```
//Q4 4. Create a structure for calender date with day, month and year.
// Find if given two days are equal, or which is earlier.
// Write a function to add days to the date structure to form the new date. Assume no leap year.
#include <iostream>
#include <string>

using namespace std;

struct date{
	int day;
	int month;
	int year;
};

void getDate(date &d)
{
	cout << endl;
	cout << "Enter day: "; cin >> d.day;
	cout << "Enter month: "; cin >> d.month;
	cout << "Enter year: "; cin >> d.year;
}

void showDate(date d)
{
	cout << endl;
	cout << "day: " << d.day;
	cout << ", month: " << d.month;
	cout << ", year: " << d.year;
}

int compareDate(date d1, date d2)
{
	if(d1.year == d2.year)
		if(d1.month == d2.month)
			if(d1.day == d2.day) return 0;
			else if(d1.day < d2.day) return -1;
			else return 1;
		else if(d1.month < d2.month) return -1;
		else return 1;
	else if(d1.year < d2.year) return -1;
	return 1;
}
```

```
date addDays(int days, date d)
{
	date newDate;
	newDate.day = d.day + days > 30 ? (d.day + days) % 30 : d.day + days;
	newDate.month = d.day + days > 30 ? d.month == 12 ? 1 : d.month + 1 : d.month;
	newDate.year = d.day + days > 30 ? d.month + 1 > 12 ? d.year + 1 : d.year : d.year;

	return newDate;
}

int main()
{
	date d1, d2;
	cout << "Enter date 1: "; getDate(d1);
	cout << "Enter date 2: "; getDate(d2);

	if(compareDate(d1, d2) == 0) cout << "Entered dates are equal\n";
	else if(compareDate(d1, d2) < 0) cout << "Day 1 comes before day 2\n";
	else cout << "Day 1 comes after day 2\n";

	int days;
	cout << "Enter number of days to add to day1: "; cin >> days;
	cout << "New date is: "; showDate(addDays(days, d1));
	return 0;
}
```

+



5. Create a structure for student with rollno, name, marks for 4 subjects. Create an list of students using array

a) calculate average for each student and print along with rollno

b) sort the array by total marks and print the same with rollno in sorted order

```
#include <iostream>
#include <string>

using namespace std;

struct student
{
   int roll;
   string name;
   int marks[4];
};

void getData(student &s, int i = 0)
{
   string show_student_number = "";
   if(i > 0) show_student_number = " " + to_string(i);
   getchar();
   cout << "Enter name of student" << show_student_number << ": "; getline(cin, s.name);
   cout << "Enter the roll number of student: "; cin >> s.roll;

   for(int i = 0; i < 4; i++)
   {
      cout << "Enter marks in subject " << i + 1 << ": "; cin >> s.marks[i];
   }
}

void showData(student s, int i = 0)
{
   string show_student_number = "";
   if(i > 0) show_student_number = " " + to_string(i);
   cout << "Name of student" << show_student_number << ": " << s.name << endl;
   cout << "Roll of student" << show_student_number << ": " << s.roll << endl;
   for(int i = 0; i < 4; i++)
   {
      cout << "Marks in subject " << i + 1 << ": " << s.marks[i] << endl;
   }
}

float getAvg(int a[], int n = 4)
```

```
{
   float sum = 0;
   for(int i = 0; i < n; i++) sum += a[i];

   return sum / n;
}

int getTotal(int a[], int n = 4)
{
   int sum = 0;
   for(int i = 0; i < n; i++) sum += a[i];

   return sum;
}

int main()
{
   int n;

   cout << "Enter number of students: "; cin >> n;

   student students[n];

   for(int i = 0; i < n; i++) getData(students[i], i + 1);

   cout << endl << "Average of each student as follows: " << endl;
   for(int i = 0; i < n; i++)
   {
      cout << "Student " << i + 1 << ":" << endl;
      cout << "Roll Number: " << students[i].roll << "; Average: " << getAvg(students[i].marks) << endl;
      cout << "----------------------------------------" << endl;
   }

   cout << endl << "Sorting the student array based on total marks..." << endl;

   for(int i = 0; i < n; i++)
   {
      for(int j = i + 1; j < n; j++)
      {
         if(getTotal(students[j - 1].marks) > getTotal(students[j].marks))
         {
            student t = students[j - 1];
            students[j - 1] = students[j];
            students[j] = t;
         }
      }
   }
```

```
    for(int i = 0; i < n; i++)
    {
        cout << "Student " << i + 1 << ":" << endl;
        cout << "Roll Number: " << students[i].roll << "; Total: " << getTotal(students[i].marks) << endl;
        cout << "---------------------------------------" << endl;
    }

    return 0;
}
```