INDUSTRIAL QUANTUM ROBOTICS
WEEK 3

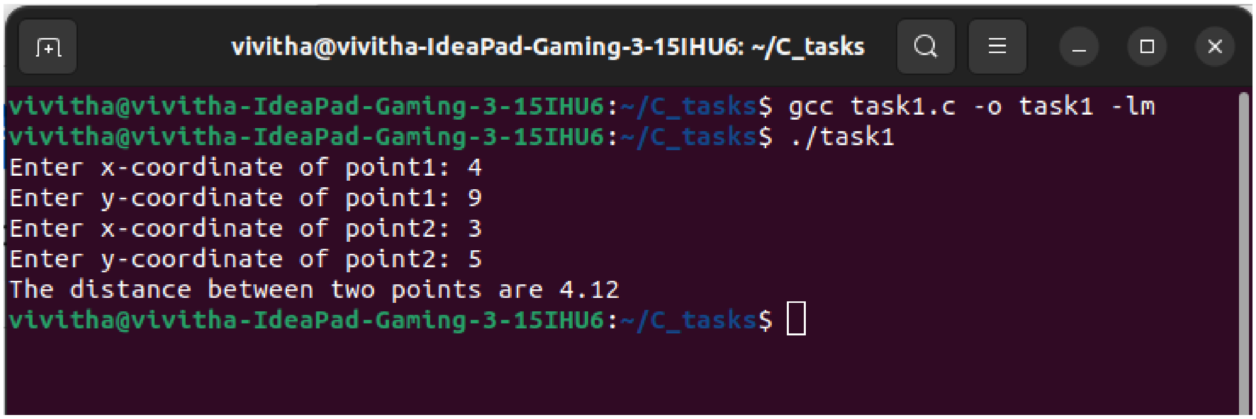
Name : Vivitha M
Roll no :7376221CS352

1. Define a structure to represent a point in 2D space (x, y). Write a function to compute the distance between two points.

CODE:

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

struct Point
{
        double x;
        double y;
};

double distance( struct Point p1, struct Point p2 )
{
        double dx = p2.x - p1.x;
        double dy = p2.y - p1.y;
        return sqrt( dx*dx + dy*dy );
}

int main() {

struct Point point1,point2;
   printf("Enter x-coordinate of point1: ");
   scanf("%lf", &point1.x);
   printf("Enter y-coordinate of point1: ");
   scanf("%lf", &point1.y);
   printf("Enter x-coordinate of point2: ");
   scanf("%lf", &point2.x);
   printf("Enter y-coordinate of point2: ");
   scanf("%lf", &point2.y);
double dist = distance(point1, point2);
printf("The distance between two points are %.2lf\n",dist);
return 0;
}
```

OUTPUT:



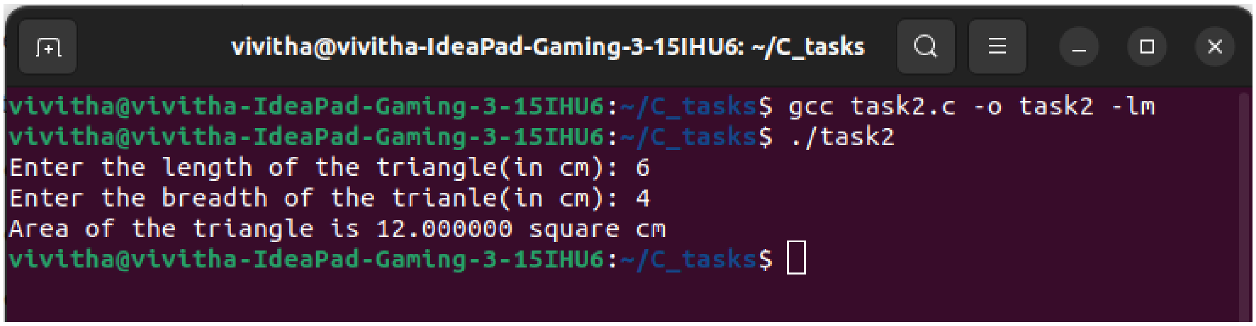
```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task1.c -o task1 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task1
Enter x-coordinate of point1: 4
Enter y-coordinate of point1: 9
Enter x-coordinate of point2: 3
Enter y-coordinate of point2: 5
The distance between two points are 4.12
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

2. Calculate the area of a rectangle using a structure to store its dimensions.

CODE:

```
#include <stdio.h>

struct dimension
{
	float breadth;
	float length;
};

float areaoftriangle(struct dimension a)
{
	return (0.5 * a.length * a.breadth) ;
}

int main() {

struct dimension Area;
printf("Enter the length of the triangle in cm: ");
scanf("%f",&Area.length);
printf("Enter the breadth of the trianle in cm: ");
scanf("%f",&Area.breadth);

float area = areaoftriangle(Area);
printf("Area of the triangle is %f square cm\n",area);
return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task2.c -o task2 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task2
Enter the length of the triangle(in cm): 6
Enter the breadth of the trianle(in cm): 4
Area of the triangle is 12.000000 square cm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

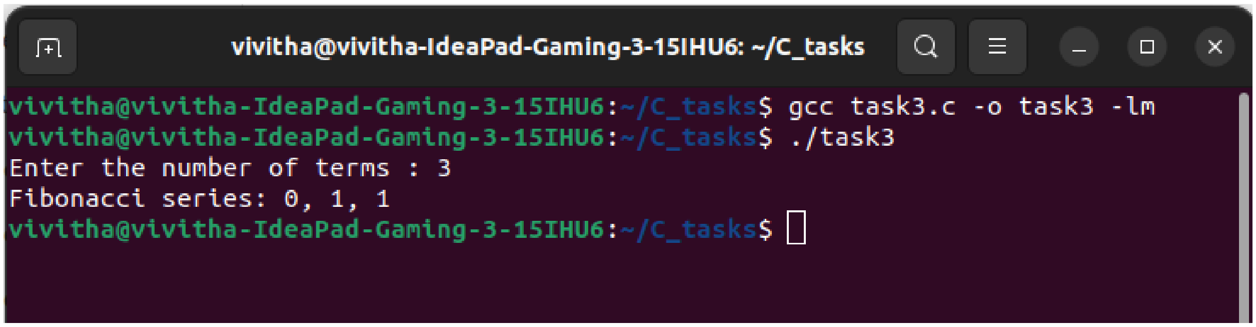
3.Write a program that generates the Fibonacci series up to a specified number of terms using structures.

CODE:

```
#include <stdio.h>
struct num {
    int number;
    int first;
    int second;
};

void Fibonacci(int n) {
    int first = 0, second = 1, next;

    printf("Fibonacci series: ");

    for (int i = 0; i < n; i++) {
        if (i <= 1) {
            next = i;
        } else {
            next = first + second;
            first = second;
            second = next;
        }

        printf("%d", next);

        if (i < n - 1) {
            printf(", ");
        }
    }

    printf("\n");
}
```

```
int main() {
    struct num num1;

    printf("Enter the number of terms in Fibonacci series: ");
    scanf("%d", &num1.number);

    if (num1.number <= 0) {
        printf("Invalid input. Number of terms must be greater than 0.\n");
        return 1;
    }

    Fibonacci(num1.number);

    return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task3.c -o task3 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task3
Enter the number of terms : 3
Fibonacci series: 0, 1, 1
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

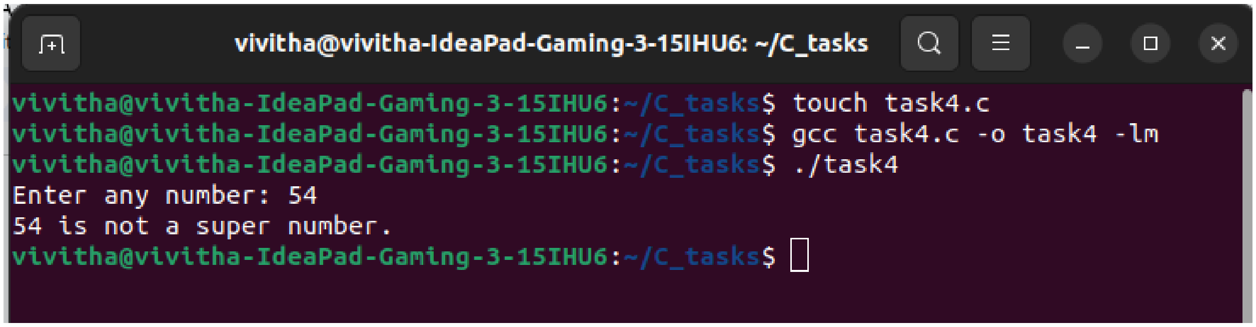
4.A program to check if a number is a super number (a number that is equal to the sum of its digits raised to the power of the number of digits).

CODE:

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

struct Number {
        int value;
        int Numdigits;
};

int countdigit(int num)
{
        int count = 0;
        while (num > 0)
        {
                num /=10;
                count++;
        }
        return count;
}

struct Number initialisenum(int value)
{
        struct Number num;
        num.value = value;
        num.Numdigits = countdigit(value);
        return num;
}

int isSupernum(struct Number num)
{
        int Originalnum = num.value;
        int temp = num.value;
        int sumOfDigits = 0;

        while (temp > 0 )
        {
                int digit = temp % 10;
                sumOfDigits += pow(digit,num.Numdigits);
                temp /=10;
        }
        return Originalnum == sumOfDigits;
}

int main() {
   int inputValue;
   struct Number num;

   printf("Enter a number: ");
   scanf("%d", &inputValue);
   num = initialisenum(inputValue);

   if (isSupernum(num)) {
      printf("%d is a super number.\n", num.value);
   } else {
      printf("%d is not a super number.\n", num.value);
   }

   return 0;
```

```
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ touch task4.c
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task4.c -o task4 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task4
Enter any number: 54
54 is not a super number.
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

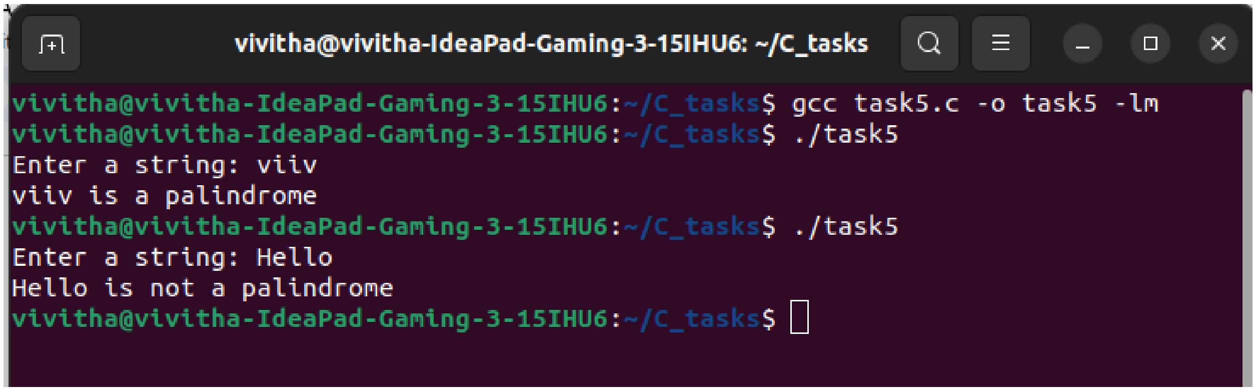
5. This program checks if a given string is a palindrome using a stack (using structure).

CODE:

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

struct palindrome {
        char string[100];
};

int checkpalindrome(struct palindrome str)
{
        int length = strlen(str.string);
        int start = 0;
        int end = length -1;

        while(start < end)
        {
        if(str.string[start] != str.string[end])
        {
                return 0;

        }
        start ++;
        end -- ;
        }
        return 1;

}

int main() {
```

```
struct palindrome input;

printf("Enter a string: ");
scanf("%s",input.string);

if(checkpalindrome(input))
{
        printf("%s is a palindrome\n",input.string);
}
else{
        printf("%s is not a palindrome\n",input.string);
}
return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task5.c -o task5 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task5
Enter a string: viiv
viiv is a palindrome
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task5
Enter a string: Hello
Hello is not a palindrome
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

6.Manage a simple bank account using a structure to store the account details.

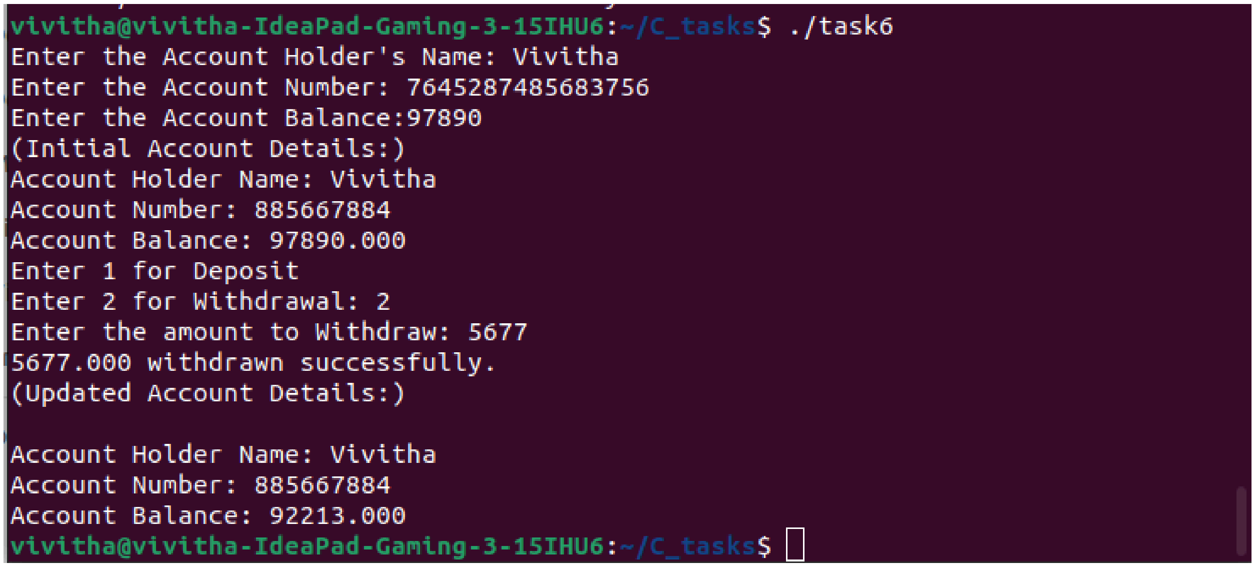
CODE:

```
#include <stdio.h>
#include <string.h>
struct bank_account {
char name[50];
int AccountNo;
double balance;
};
void displayAccount(struct bank_account account)
{
printf("Account Holder Name: %s\n", account.name);
printf("Account Number: %d\n", account.AccountNo);
printf("Account Balance: %.3lf\n", account.balance);
```

```
}
struct bank_account deposit(struct bank_account account, double amount)
{
if (amount > 0) {
account.balance += amount;
printf("%.3lf deposited successfully.\n", amount);
}
else {
printf("Invalid deposit amount.\n");
}
return account;
}
struct bank_account withdraw(struct bank_account account, double amount)
{
if (amount > 0 && account.balance >= amount) {
account.balance -= amount;
printf("%.3lf withdrawn successfully.\n", amount);
}
else {
printf("Invalid withdraw amount or insufficient balance.\n");
}
return account;
}
int main() {
double amount;
struct bank_account myaccount;
printf("Enter the Account Holder's Name: ");
scanf("%s", myaccount.name);
printf("Enter the Account Number: ");
scanf("%d", &myaccount.AccountNo);
printf("Enter the Account Balance:");
scanf("%lf", &myaccount.balance);
printf("(Initial Account Details:)\n");
displayAccount(myaccount);
printf("Enter 1 for Deposit\nEnter 2 for Withdrawal: ");
int choice;
scanf("%d", &choice);
if (choice == 1) {
printf("Enter the amount to Deposit: ");
scanf("%lf", &amount);
myaccount = deposit(myaccount, amount);
}
else if (choice == 2) {
```

```
printf("Enter the amount to Withdraw: ");
scanf("%lf", &amount);
myaccount = withdraw(myaccount, amount);
}
else {
printf("Please choose a valid option\n");
}
printf("(Updated Account Details:)\n\n");
displayAccount(myaccount);
return 0;
}
```

OUTPUT:



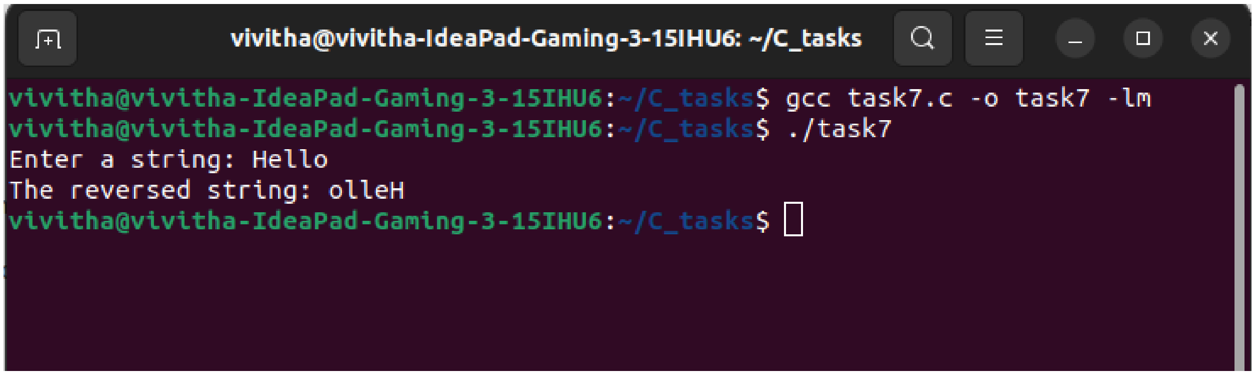
```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task6
Enter the Account Holder's Name: Vivitha
Enter the Account Number: 7645287485683756
Enter the Account Balance:97890
(Initial Account Details:)
Account Holder Name: Vivitha
Account Number: 885667884
Account Balance: 97890.000
Enter 1 for Deposit
Enter 2 for Withdrawal: 2
Enter the amount to Withdraw: 5677
5677.000 withdrawn successfully.
(Updated Account Details:)

Account Holder Name: Vivitha
Account Number: 885667884
Account Balance: 92213.000
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

7.Write a program to reverse an array using function

CODE:

```
#include <stdio.h>
#include <string.h>
struct string {
        char str[100];
};
struct string reversestring(struct string revstr)
{
        int length = strlen(revstr.str);
        int start = 0;
        int end = length -1;
        char temp;
        while(start < end)
        {
                temp = revstr.str[start];
                revstr.str[start] = revstr.str[end];
                revstr.str[end] = temp;
                start ++;
                end --;
        }
        return revstr;
}
int main() {
struct string input;
printf("Enter a string: ");
scanf("%s",input.str);
input = reversestring(input);
printf("The reversed string: %s\n",input.str);
return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task7.c -o task7 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task7
Enter a string: Hello
The reversed string: olleH
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

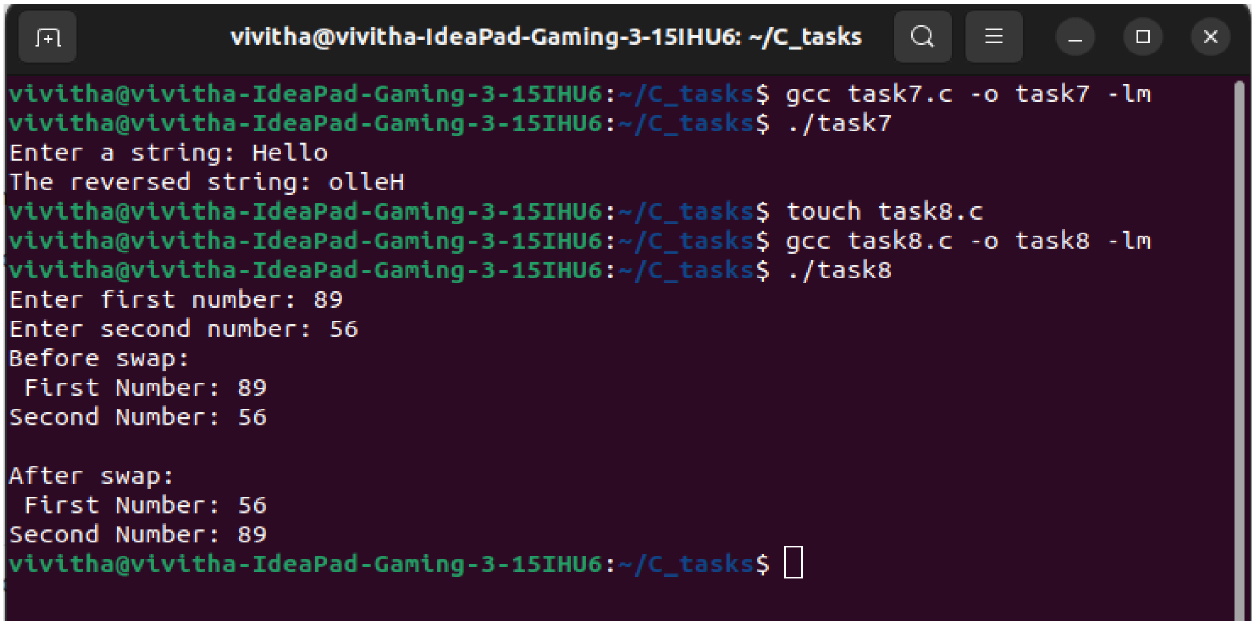
8. This program swaps the values of two numbers using function.

CODE:

```
#include <stdio.h>
struct swap {
        int number1,number2;
};
struct swap swapvar(struct swap nums)
{
        int temp;
        temp = nums.number1;
        nums.number1 = nums.number2;
        nums.number2 = temp;
        return nums;
}
int main() {
```

```
struct swap numbers;
printf("Enter first number: ");
scanf("%d",&numbers.number1);
printf("Enter second number: ");
scanf("%d",&numbers.number2);
printf("Before swap:\n ");
printf("First Number: %d\nSecond Number:
%d\n\n",numbers.number1,numbers.number2);
printf("After swap:\n ");
numbers = swapvar(numbers);
printf("First Number: %d\nSecond Number:
%d\n",numbers.number1,numbers.number2);
return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task7.c -o task7 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task7
Enter a string: Hello
The reversed string: olleH
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ touch task8.c
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task8.c -o task8 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task8
Enter first number: 89
Enter second number: 56
Before swap:
 First Number: 89
Second Number: 56

After swap:
 First Number: 56
Second Number: 89
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

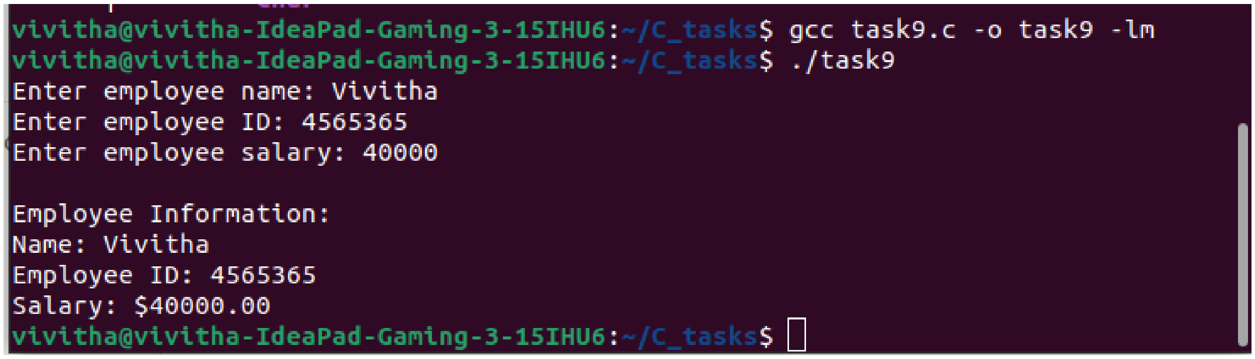
9.This program defines a structure to store employee information (name, employee ID, and salary) and displays it.

CODE:

```
#include <stdio.h>
struct Employee {
        char name[100];
        int employeeID;
        double salary;
};
```

```
int main() {
        struct Employee employee;
        printf("Enter employee name: ");
        scanf("%s", employee.name);

        printf("Enter employee ID: ");
        scanf("%d", &employee.employeeID);

        printf("Enter employee salary: ");
        scanf("%lf", &employee.salary);
        printf("\nEmployee Information:\n");
        printf("Name: %s\n", employee.name);
        printf("Employee ID: %d\n", employee.employeeID);
        printf("Salary: $%.2lf\n", employee.salary);

        return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task9.c -o task9 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task9
Enter employee name: Vivitha
Enter employee ID: 4565365
Enter employee salary: 40000

Employee Information:
Name: Vivitha
Employee ID: 4565365
Salary: $40000.00
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

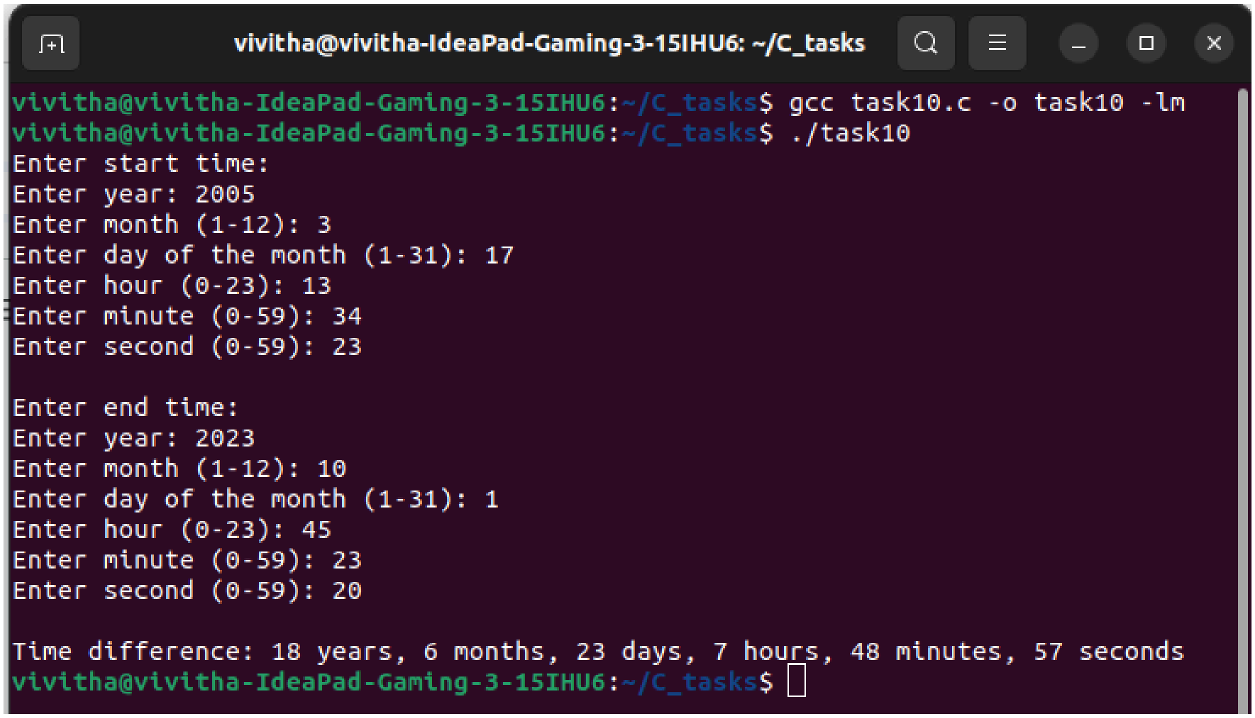
10.Calculate the difference between two time periods.

CODE:

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

struct Time {
        int year;
        int month;
        int day;
        int hour;
        int minute;
        int second;
```

```
};

struct Time inputTime() {
    struct Time t;
    printf("Enter year: ");
    scanf("%d", &t.year);
    printf("Enter month (1-12): ");
    scanf("%d", &t.month);
    printf("Enter day of the month (1-31): ");
    scanf("%d", &t.day);
 printf("Enter hour (0-23): ");
    scanf("%d", &t.hour);
    printf("Enter minute (0-59): ");
    scanf("%d", &t.minute);
    printf("Enter second (0-59): ");
    scanf("%d", &t.second);
    return t;
}

struct Time calculateTimeDifference(struct Time start, struct Time end) {
    struct Time diff = {0, 0, 0, 0, 0, 0};

    struct tm start_time = {
    .tm_year = start.year - 1900,
    .tm_mon = start.month - 1,
    .tm_mday = start.day,
    .tm_hour = start.hour,
    .tm_min = start.minute,
    .tm_sec = start.second
    };

    struct tm end_time = {
    .tm_year = end.year - 1900,
    .tm_mon = end.month - 1,
    .tm_mday = end.day,
    .tm_hour = end.hour,
    .tm_min = end.minute,
    .tm_sec = end.second
    };

    time_t start_seconds = mktime(&start_time);
    time_t end_seconds = mktime(&end_time);
    time_t difference = end_seconds - start_seconds;

    diff.year = difference / 31536000;
    difference -= diff.year * 31536000;
    diff.month = difference / 2592000;
    difference -= diff.month * 2592000;
    diff.day = difference / 86400;
    difference -= diff.day * 86400;
    diff.hour = difference / 3600;
    difference -= diff.hour * 3600;
    diff.minute = difference / 60;
    diff.second = difference % 60;

    return diff;
}

int main() {
    struct Time startTime, endTime, timeDifference;

    printf("Enter start time:\n");
    startTime = inputTime();

    printf("\nEnter end time:\n");
    endTime = inputTime();

    timeDifference = calculateTimeDifference(startTime, endTime);

    printf("\nTime difference: %d years, %d months, %d days, %d hours, %d minutes,
%d seconds\n",
    timeDifference.year, timeDifference.month, timeDifference.day,
  timeDifference.hour, timeDifference.minute, timeDifference.second);

    return 0;
}
```

OUTPUT:



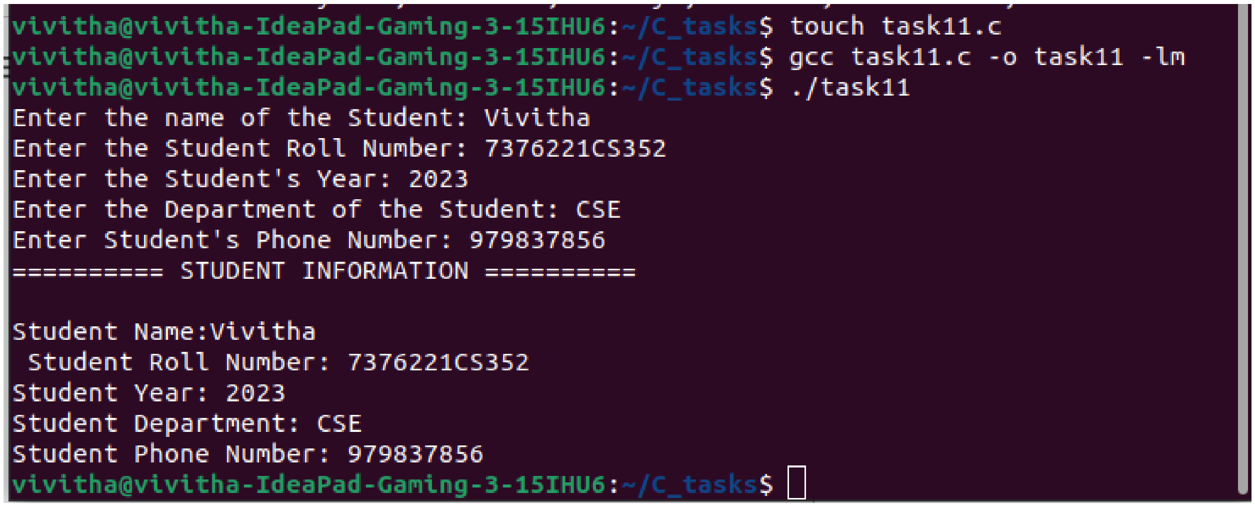
```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task10.c -o task10 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task10
Enter start time:
Enter year: 2005
Enter month (1-12): 3
Enter day of the month (1-31): 17
Enter hour (0-23): 13
Enter minute (0-59): 34
Enter second (0-59): 23

Enter end time:
Enter year: 2023
Enter month (1-12): 10
Enter day of the month (1-31): 1
Enter hour (0-23): 45
Enter minute (0-59): 23
Enter second (0-59): 20

Time difference: 18 years, 6 months, 23 days, 7 hours, 48 minutes, 57 seconds
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

11. C Program to Store Information of a Student Using Structure.

CODE:

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

struct std {
   char name[50];
   char Id[50];
   int year;
   char dept[50];
   char phone[15];
};

void information(struct std student)
{
   printf("========== STUDENT INFORMATION ==========\n\n");
   printf("Student Name:%s\n ",student.name);
   printf("Student Roll Number: %s\n",student.Id);
   printf("Student Year: %d\n",student.year);
   printf("Student Department: %s\n",student.dept);
   printf("Student Phone Number: %s\n",student.phone);
}

int main() {

struct std input;

printf("Enter the name of the Student: ");
scanf("%s",input.name);

printf("Enter the Student Roll Number: ");
scanf("%s",input.Id);

printf("Enter the Student's Year: ");
scanf("%d",&input.year);

printf("Enter the Department of the Student: ");
scanf("%s",input.dept);

printf("Enter Student's Phone Number: ");
scanf("%s",input.phone);
information(input);
return 0;

}
```

OUTPUT:



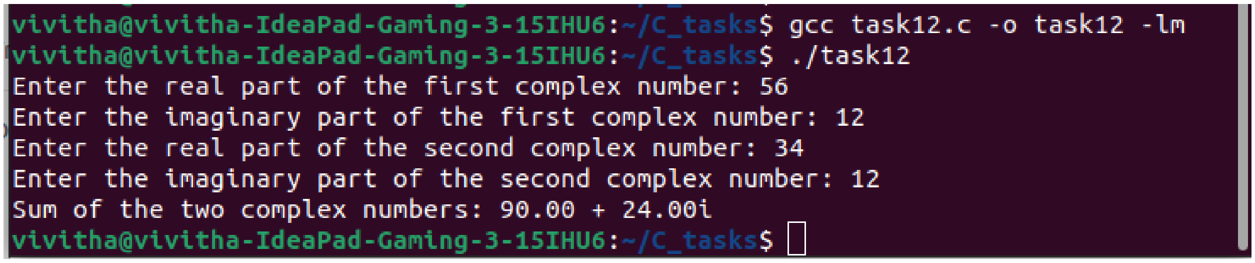
```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ touch task11.c
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task11.c -o task11 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task11
Enter the name of the Student: Vivitha
Enter the Student Roll Number: 7376221CS352
Enter the Student's Year: 2023
Enter the Department of the Student: CSE
Enter Student's Phone Number: 979837856
========== STUDENT INFORMATION ==========

Student Name:Vivitha
 Student Roll Number: 7376221CS352
Student Year: 2023
Student Department: CSE
Student Phone Number: 979837856
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

12 Add Two Complex Numbers by Passing Structure to a Function.

CODE:

```
#include <stdio.h>

struct Complex {
        double real;
        double imag;
};

struct Complex addComplex(struct Complex num1, struct Complex num2) {
        struct Complex result;
        result.real = num1.real + num2.real;
        result.imag = num1.imag + num2.imag;
        return result;
}

int main() {
        struct Complex complex1, complex2, sum;

        printf("Enter the real part of the first complex number: ");
        scanf("%lf", &complex1.real);
        printf("Enter the imaginary part of the first complex number: ");
        scanf("%lf", &complex1.imag);

        printf("Enter the real part of the second complex number: ");
        scanf("%lf", &complex2.real);
        printf("Enter the imaginary part of the second complex number: ");
        scanf("%lf", &complex2.imag);
        sum = addComplex(complex1, complex2);
        printf("Sum of the two complex numbers: %.2lf + %.2lfi\n", sum.real, sum.imag);

        return 0;
}
```

OUTPUT:



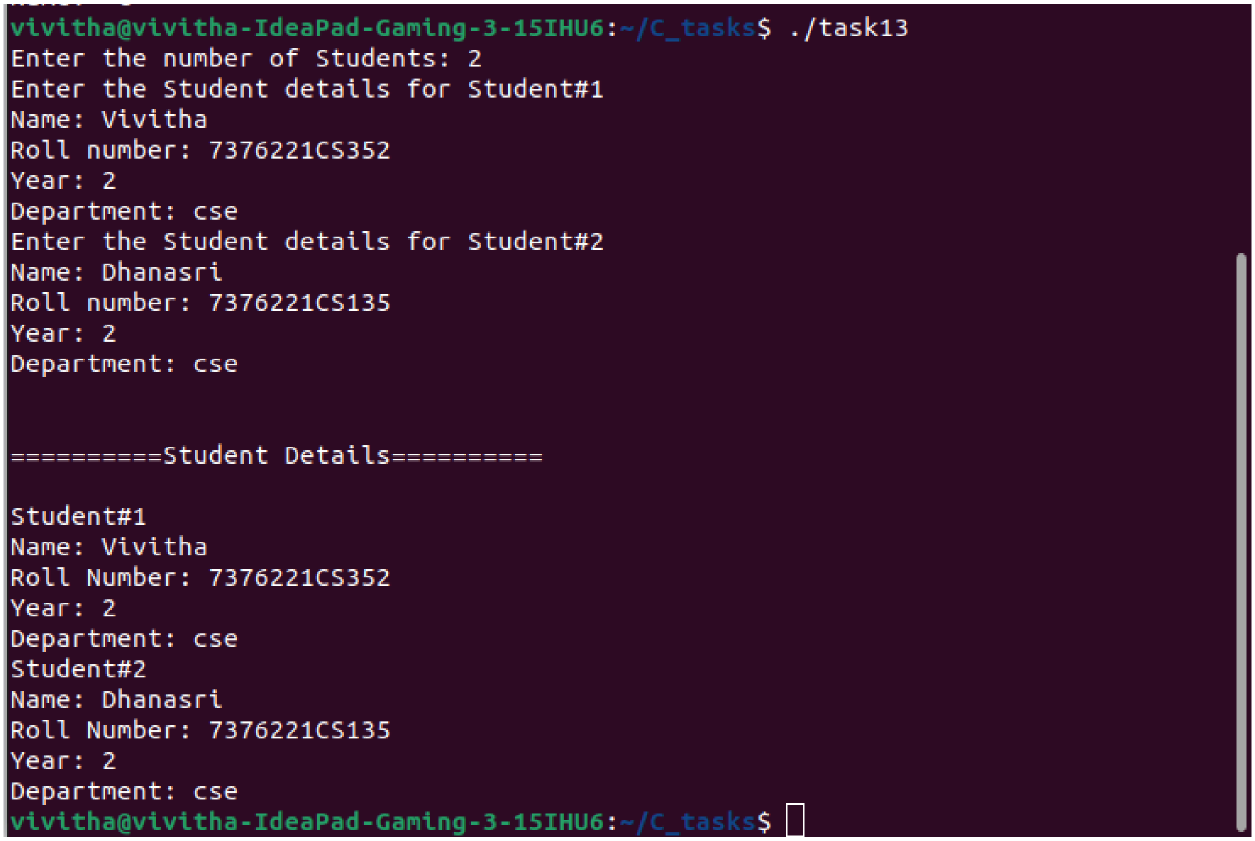
```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task12.c -o task12 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task12
Enter the real part of the first complex number: 56
Enter the imaginary part of the first complex number: 12
Enter the real part of the second complex number: 34
Enter the imaginary part of the second complex number: 12
Sum of the two complex numbers: 90.00 + 24.00i
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

13.Store information of n students using structures

CODE:

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


struct student {

        char name[50];
        char rollno[50];
        int year;
        char dept[50];

};
struct student n_students(struct student std[], int n)
{

        for(int i=0; i<n ; i++)
        {
        printf("Enter the Student details for Student#%d\n",i+1);
        printf("Name: ");
        scanf("%s",std[i].name);

        printf("Roll number: ");
        scanf("%s",std[i].rollno);

        printf("Year: ");
        scanf("%d",&std[i].year);

        printf("Department: ");
        scanf("%s",std[i].dept);

        }

        printf("\n\n==========Student Details==========\n\n");

        for(int i=0; i<n; i++)
        {
        printf("Student#%d\n",i+1);
        printf("Name: %s\n",std[i].name);
        printf("Roll Number: %s\n",std[i].rollno);
        printf("Year: %d\n",std[i].year);
        printf("Department: %s\n",std[i].dept);

        }
}

int main() {
int number;
printf("Enter the number of Students: ");
scanf("%d",&number);
struct student Std[number];
n_students(Std,number);
return 0;

}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task13
Enter the number of Students: 2
Enter the Student details for Student#1
Name: Vivitha
Roll number: 7376221CS352
Year: 2
Department: cse
Enter the Student details for Student#2
Name: Dhanasri
Roll number: 7376221CS135
Year: 2
Department: cse


==========Student Details==========

Student#1
Name: Vivitha
Roll Number: 7376221CS352
Year: 2
Department: cse
Student#2
Name: Dhanasri
Roll Number: 7376221CS135
Year: 2
Department: cse
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$
```

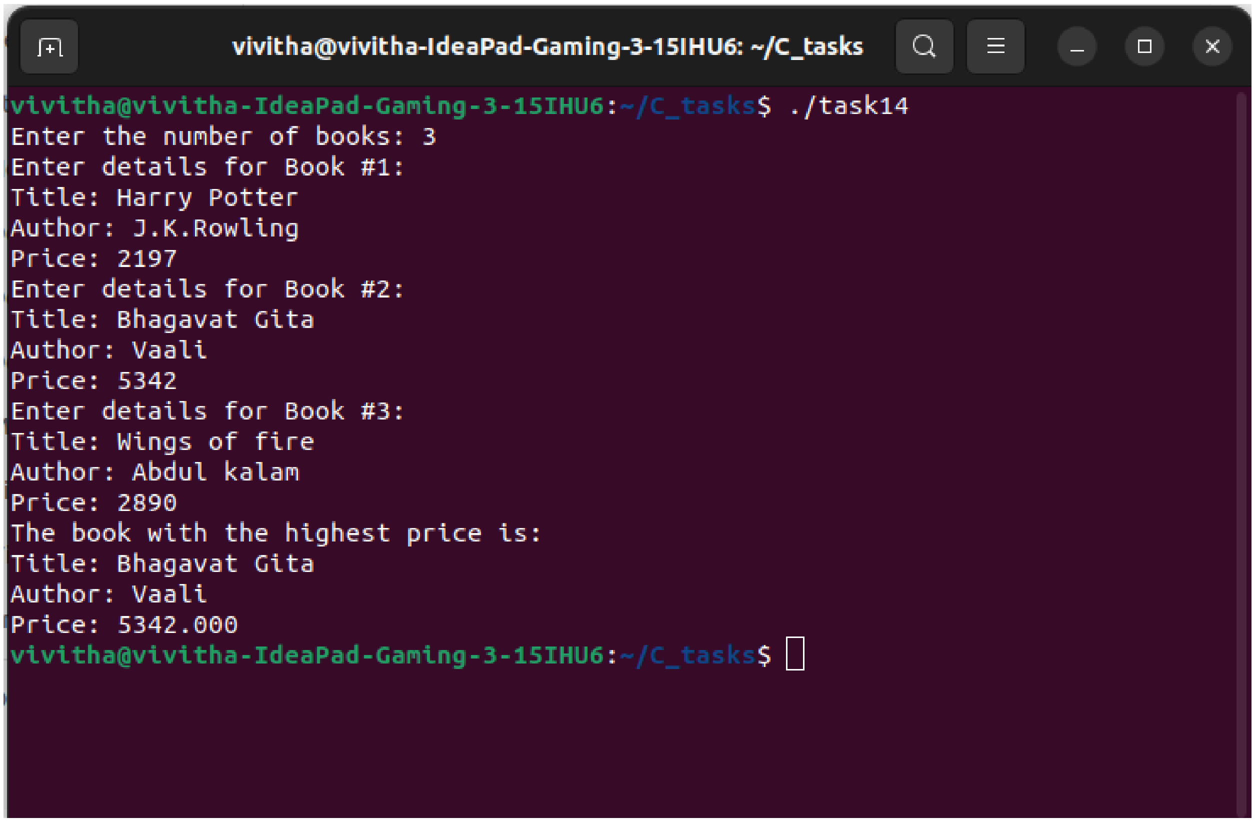
14.Define a structure called Book with members for title, author, and price. Write a function that takes an array of books as input and returns the book with the highest price.

CODE:

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

```
#include <string.h>
struct Book {
char title[100];
char author[100];
float price;
};
struct Book findHighestPricedBook(struct Book books[], int numBooks) {
struct Book highestPricedBook = books[0];
for (int i = 1; i < numBooks; i++) {
if (books[i].price > highestPricedBook.price) {
highestPricedBook = books[i];
}
}
return highestPricedBook;
}
int main() {
int numBooks;
printf("Enter the number of books: ");
scanf("%d", &numBooks);
struct Book books[numBooks];
for (int i = 0; i < numBooks; i++) {
printf("Enter details for Book #%d:\n", i + 1);
printf("Title: ");
scanf(" %[^\n]", books[i].title);
printf("Author: ");
scanf(" %[^\n]", books[i].author);
printf("Price: ");
scanf("%f", &books[i].price);
}
struct Book highestPriceBook = findHighestPricedBook(books, numBooks);
printf("The book with the highest price is:\n");
printf("Title: %s\n", highestPriceBook.title);
printf("Author: %s\n", highestPriceBook.author);
printf("Price: %.3f\n", highestPriceBook.price);
return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task14
Enter the number of books: 3
Enter details for Book #1:
Title: Harry Potter
Author: J.K.Rowling
Price: 2197
Enter details for Book #2:
Title: Bhagavat Gita
Author: Vaali
Price: 5342
Enter details for Book #3:
Title: Wings of fire
Author: Abdul kalam
Price: 2890
The book with the highest price is:
Title: Bhagavat Gita
Author: Vaali
Price: 5342.000
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

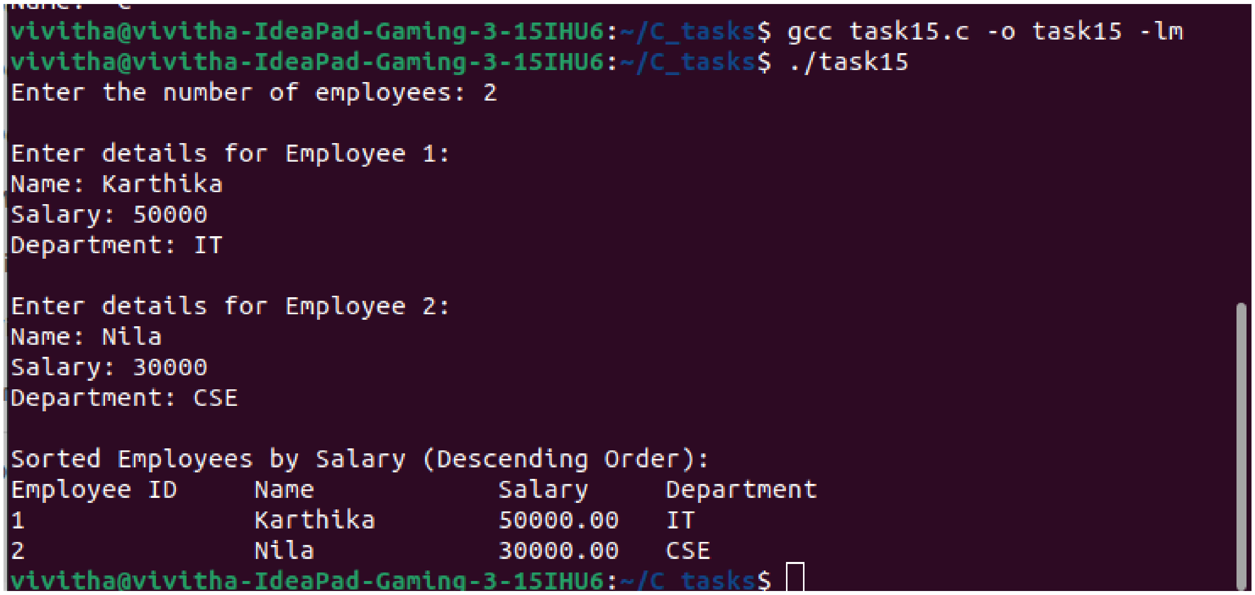
15.Create a structure named Employee with members for employee_id, name, salary, and department. Write a program that reads data for a list of employees, stores it in an array of structures, and then sorts the employees based on their salaries in descending order.

CODE:

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




struct Employee {
int employee_id;
char name[50];
float salary;
char department[50];
};


void swap(struct Employee a, struct Employee b) {
struct Employee temp = a;
```

```
a = b;
b = temp;
}




int main() {
int n;


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


struct Employee employees[n];




for (int i = 0; i < n; i++) {
printf("\nEnter details for Employee %d:\n", i + 1);
employees[i].employee_id = i + 1;


printf("Name: ");
scanf("%s", employees[i].name);


printf("Salary: ");
scanf("%f", &employees[i].salary);


printf("Department: ");
scanf("%s", employees[i].department);
}

for (int i = 0; i < n - 1; i++) {
for (int j = 0; j < n - i - 1; j++) {
if (employees[j].salary < employees[j + 1].salary) {
swap(employees[j], employees[j + 1]);
      }
   }
}
printf("\nSorted Employees by Salary (Descending Order):\n");
printf("%-15s %-15s %-10s %-15s\n", "Employee ID", "Name", "Salary",
"Department");
for (int i = 0; i < n; i++) {
printf("%-15d %-15s %-10.2f %-15s\n", employees[i].employee_id, employees[i].name,
employees[i].salary, employees[i].department);
}


return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task15.c -o task15 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task15
Enter the number of employees: 2

Enter details for Employee 1:
Name: Karthika
Salary: 50000
Department: IT

Enter details for Employee 2:
Name: Nila
Salary: 30000
Department: CSE

Sorted Employees by Salary (Descending Order):
Employee ID     Name            Salary     Department
1               Karthika        50000.00   IT
2               Nila            30000.00   CSE
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$
```

16.Define a structure named Point to represent a point in 3D space with members for x, y, and z coordinates. Write a function to calculate the distance between two points in 3D space using the Euclidean distance formula.

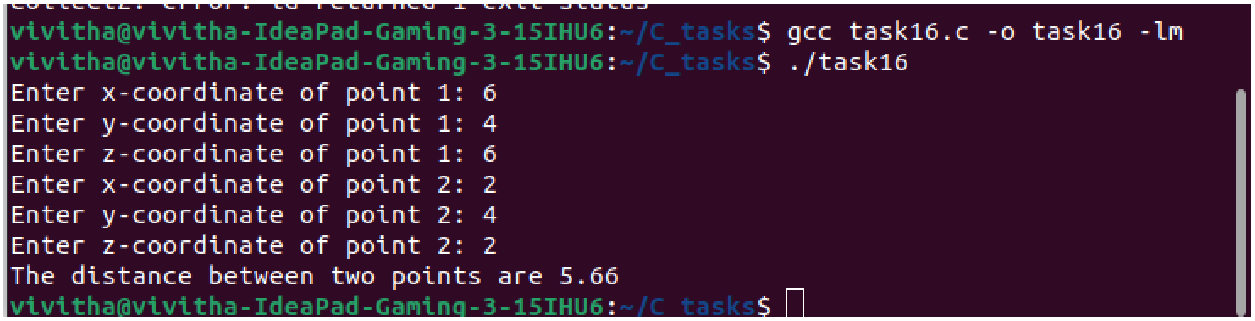
CODE:

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


struct dist{

        double x;
        double y;
        double z;
```

```
};


double distance(struct dist p1, struct dist p2)
{
        double dx = p2.x - p1.x;
        double dy = p2.y - p1.y;
        double dz = p2.z - p1.z;

        return sqrt(dx*dx + dy*dy + dz*dz);

}


int main() {


struct dist point1, point2;


printf("Enter x-coordinate of point 1: ");
scanf("%lf", &point1.x);


printf("Enter y-coordinate of point 1: ");
scanf("%lf", &point1.y);


printf("Enter z-coordinate of point 1: ");
scanf("%lf", &point1.z);


printf("Enter x-coordinate of point 2: ");
scanf("%lf", &point2.x);


printf("Enter y-coordinate of point 2: ");
scanf("%lf", &point2.y);


printf("Enter z-coordinate of point 2: ");
scanf("%lf", &point2.z);
```

```
double total = distance(point1,point2);


printf("The distance between two points are %.2lf\n",total);


return 0;


}
```

OUTPUT:



```
collect2: error: ld returned 1 exit status
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task16.c -o task16 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task16
Enter x-coordinate of point 1: 6
Enter y-coordinate of point 1: 4
Enter z-coordinate of point 1: 6
Enter x-coordinate of point 2: 2
Enter y-coordinate of point 2: 4
Enter z-coordinate of point 2: 2
The distance between two points are 5.66
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

17.Implement a program that simulates a simple banking system. Create a structure for Account with members for account_number, account_holder_name, balance, and account_type (e.g., savings or checking). Write functions to deposit, withdraw, and transfer money between accounts while ensuring that the balance is updated correctly.

CODE:

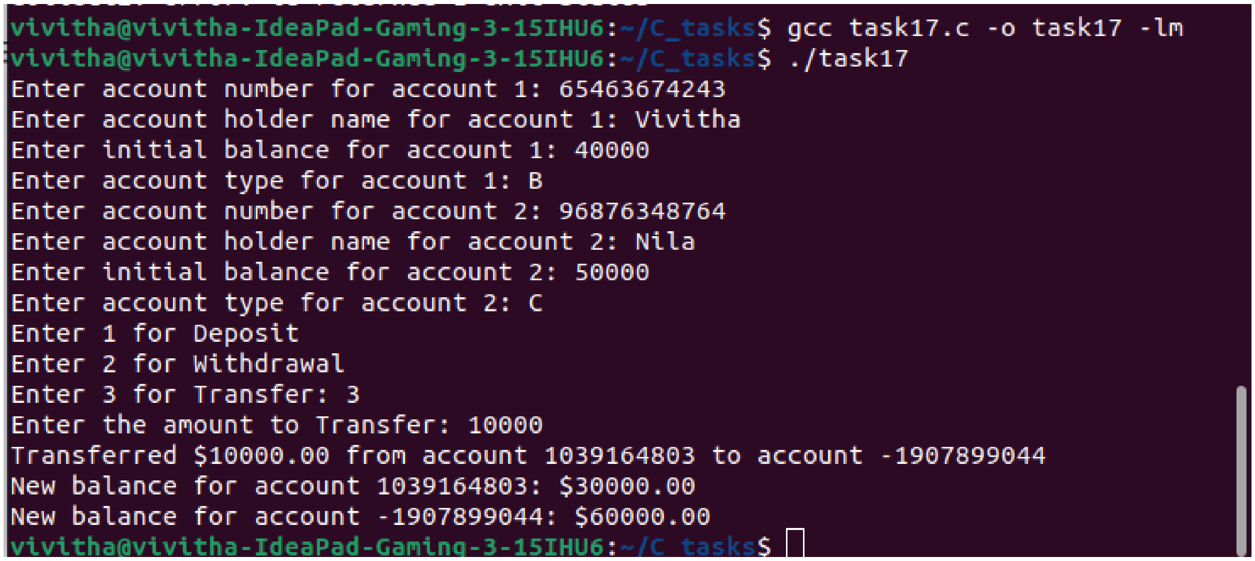
```
#include <stdio.h>
#include <string.h>


struct Account {
int account_number;
char account_holder_name[100];
double balance;
char account_type[20];
};


struct Account deposit(struct Account account, double amount) {
if (amount > 0) {
```

```
account.balance += amount;
printf("Deposited $%.2lf. New balance: $%.2lf\n",
amount, account.balance);
} else {
printf("Invalid deposit amount. Amount must be greater than 0.\n");
}
return account;
}


struct Account withdraw(struct Account account, double amount) {
if (amount > 0 && amount <= account.balance) {
account.balance -= amount;
printf("Withdrew $%.2lf. New balance: $%.2lf\n",
amount, account.balance);
} else {
printf("Invalid withdrawal amount or insufficient funds.\n");
}
return account;
}


struct Account transfer(struct Account from_account, struct Account to_account, double
amount) {
if (amount > 0 && amount <= from_account.balance) {
from_account.balance -= amount;
to_account.balance += amount;
printf("Transferred $%.2lf from account %d to account %d\n",
amount, from_account.account_number, to_account.account_number);
printf("New balance for account %d: $%.2lf\n", from_account.account_number,
from_account.balance);
printf("New balance for account %d: $%.2lf\n", to_account.account_number,
to_account.balance);
} else {
printf("Invalid transfer amount or insufficient funds.\n");
}
return from_account;
}


int main() {
struct Account myaccount1;
printf("Enter account number for account 1: ");
```

```
scanf("%d", &myaccount1.account_number);
printf("Enter account holder name for account 1: ");
scanf("%s", myaccount1.account_holder_name);
printf("Enter initial balance for account 1: ");
scanf("%lf", &myaccount1.balance);
printf("Enter account type for account 1: ");
scanf("%s", myaccount1.account_type);


struct Account myaccount2;
printf("Enter account number for account 2: ");
scanf("%d", &myaccount2.account_number);
printf("Enter account holder name for account 2: ");
scanf("%s", myaccount2.account_holder_name);
printf("Enter initial balance for account 2: ");
scanf("%lf", &myaccount2.balance);
printf("Enter account type for account 2: ");
scanf("%s", myaccount2.account_type);


int choice;
double amount;


printf("Enter 1 for Deposit\nEnter 2 for Withdrawal\nEnter 3 for Transfer: ");
scanf("%d", &choice);


if (choice == 1) {
printf("Enter the amount to Deposit: ");
scanf("%lf", &amount);
myaccount1 = deposit(myaccount1, amount);
}
else if (choice == 2) {
printf("Enter the amount to Withdraw: ");
scanf("%lf", &amount);
myaccount1 = withdraw(myaccount1, amount);
}
else if (choice == 3) {
printf("Enter the amount to Transfer: ");
scanf("%lf", &amount);
myaccount1 = transfer(myaccount1, myaccount2, amount);
}
```

```
else {
printf("Please choose a valid option\n");
}


return 0;
}
```

OUTPUT:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task17.c -o task17 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task17
Enter account number for account 1: 65463674243
Enter account holder name for account 1: Vivitha
Enter initial balance for account 1: 40000
Enter account type for account 1: B
Enter account number for account 2: 96876348764
Enter account holder name for account 2: Nila
Enter initial balance for account 2: 50000
Enter account type for account 2: C
Enter 1 for Deposit
Enter 2 for Withdrawal
Enter 3 for Transfer: 3
Enter the amount to Transfer: 10000
Transferred $10000.00 from account 1039164803 to account -1907899044
New balance for account 1039164803: $30000.00
New balance for account -1907899044: $60000.00
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

18. Create a structure called InventoryItem with members for product_code, description, unit_price, and quantity_in_stock. Write a program that allows the user to perform various inventory operations, such as adding new items, updating item details, and displaying the total value of the inventory.

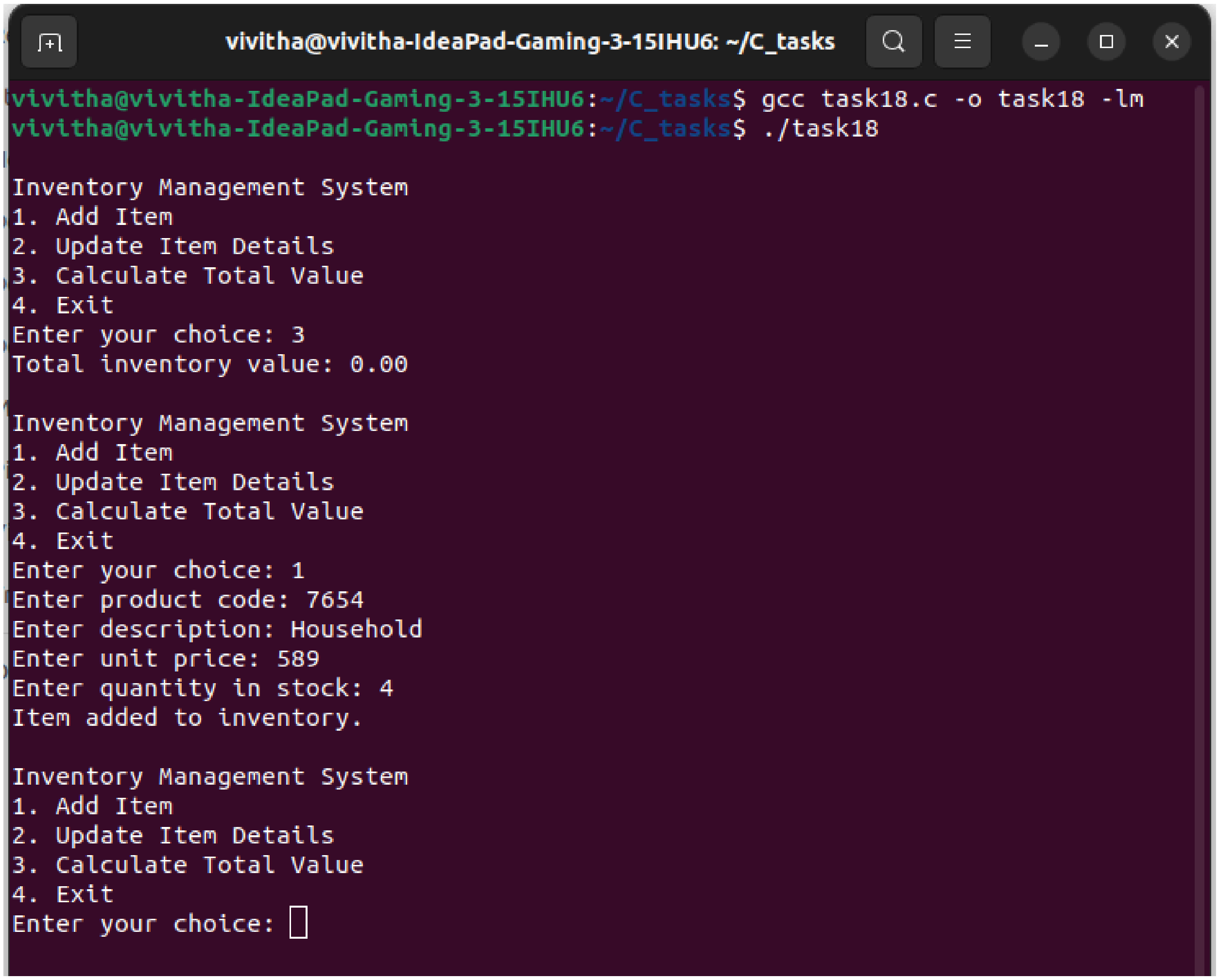
Code:

```
#include <stdio.h>
#include <string.h>
struct InventoryItem {
        int product_code;
        char description[100];
        double unit_price;
        int quantity_in_stock;
};
void addItem(struct InventoryItem inventory[], int *itemCount) {
        if (*itemCount < 100) {
        struct InventoryItem newItem;
        printf("Enter product code: ");
```

```
    scanf("%d", &newItem.product_code);
    printf("Enter description: ");
    scanf(" %[^\n]", newItem.description);
    printf("Enter unit price: ");
    scanf("%lf", &newItem.unit_price);
    printf("Enter quantity in stock: ");
    scanf("%d", &newItem.quantity_in_stock);

    inventory[*itemCount] = newItem;
    (*itemCount)++;
    printf("Item added to inventory.\n");
    } else {
    printf("Inventory is full. Cannot add more items.\n");
    }
}
void updateItem(struct InventoryItem inventory[], int itemCount) {
    int code;
    printf("Enter the product code of the item to update: ");
    scanf("%d", &code);

    int found = 0;
    for (int i = 0; i < itemCount; i++) {
    if (inventory[i].product_code == code) {
    printf("Enter new unit price: ");
    scanf("%lf", &inventory[i].unit_price);
    printf("Enter new quantity in stock: ");
    scanf("%d", &inventory[i].quantity_in_stock);
    printf("Item details updated.\n");
    found = 1;
    break;
    }
    }

    if (!found) {
    printf("Item with product code %d not found.\n", code);
    }
}
```

```
double calculateTotalValue(struct InventoryItem inventory[], int itemCount) {
    double totalValue = 0.0;
    for (int i = 0; i < itemCount; i++) {
    totalValue += inventory[i].unit_price * inventory[i].quantity_in_stock;
    }
    return totalValue;
}

int main() {
    struct InventoryItem inventory[100];
    int itemCount = 0;
    int choice;

    while (1) {
    printf("\nInventory Management System\n");
    printf("1. Add Item\n");
    printf("2. Update Item Details\n");
    printf("3. Calculate Total Value\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
    case 1:
        addItem(inventory, &itemCount);
        break;
    case 2:
        updateItem(inventory, itemCount);
        break;
    case 3:
        printf("Total inventory value: %.2lf\n", calculateTotalValue(inventory,
itemCount));
        break;
    case 4:
        return 0;
    default:
        printf("Invalid choice. Please try again.\n");
    }
    }

    return 0;
}
```

Output:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task18.c -o task18 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task18

Inventory Management System
1. Add Item
2. Update Item Details
3. Calculate Total Value
4. Exit
Enter your choice: 3
Total inventory value: 0.00

Inventory Management System
1. Add Item
2. Update Item Details
3. Calculate Total Value
4. Exit
Enter your choice: 1
Enter product code: 7654
Enter description: Household
Enter unit price: 589
Enter quantity in stock: 4
Item added to inventory.

Inventory Management System
1. Add Item
2. Update Item Details
3. Calculate Total Value
4. Exit
Enter your choice: 
```

19.Create a structure named Date with members for day, month, and year. Write functions to perform the following operations:

- Initialize a date structure with a given day, month, and year.
- Display the date in a user-friendly format (e.g., "January 1, 2023").
- Calculate the number of days between two dates.
- Determine if a given year is a leap year or not.

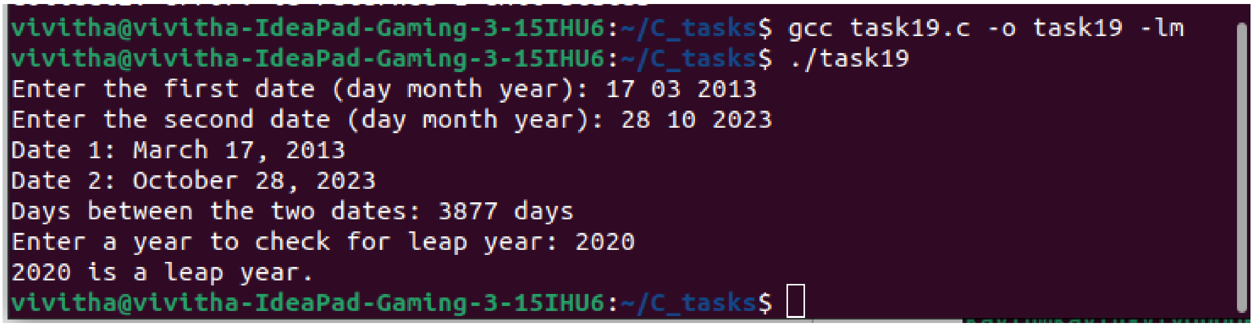
Code:

```
#include <stdio.h>
struct Date {
    int day;
    int month;
    int year;
};
int isLeapYear(int year) {
    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}
int daysInMonth(int month, int year) {
    int days[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (month == 2 && isLeapYear(year)) {
    return 29;
    }
    return days[month];
}
struct Date createDate(int day, int month, int year) {
    struct Date date;
    date.day = day;
date.month = month;
    date.year = year;
    return date;
}
void format(struct Date date, char formattedDate[20]) {
    char monthNames[][15] = {
    "January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"
    };
    sprintf(formattedDate, "%s %d, %d", monthNames[date.month - 1], date.day,
date.year);
}

int daysBetween(struct Date date1, struct Date date2) {
    int days = 0;
```

```
        while (date1.year != date2.year || date1.month != date2.month || date1.day !=
date2.day) {
        days++;
        date1.day++;
        if (date1.day > daysInMonth(date1.month, date1.year)) {
        date1.day = 1;
        date1.month++;
        if (date1.month > 12) {
                date1.month = 1;
                date1.year++;
        }
        }
        }
        return days;
}
int main() {
        struct Date date1, date2;
        printf("Enter the first date (day month year): ");
        scanf("%d %d %d", &date1.day, &date1.month, &date1.year);
        printf("Enter the second date (day month year): ");
        scanf("%d %d %d", &date2.day, &date2.month, &date2.year);
        char formattedDate1[20], formattedDate2[20];
        format(date1, formattedDate1);
        format(date2, formattedDate2);
        printf("Date 1: %s\n", formattedDate1);
        printf("Date 2: %s\n", formattedDate2);
        int daysDifference = daysBetween(date1, date2);
        printf("Days between the two dates: %d days\n", daysDifference);
        int yearToCheck;
        printf("Enter a year to check for leap year: ");
        scanf("%d", &yearToCheck);
        if (isLeapYear(yearToCheck)) {
        printf("%d is a leap year.\n", yearToCheck);
        } else {
        printf("%d is not a leap year.\n", yearToCheck);
        }
        return 0;
```

```
}
```

Output:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task19.c -o task19 -lm
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task19
Enter the first date (day month year): 17 03 2013
Enter the second date (day month year): 28 10 2023
Date 1: March 17, 2013
Date 2: October 28, 2023
Days between the two dates: 3877 days
Enter a year to check for leap year: 2020
2020 is a leap year.
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ 
```

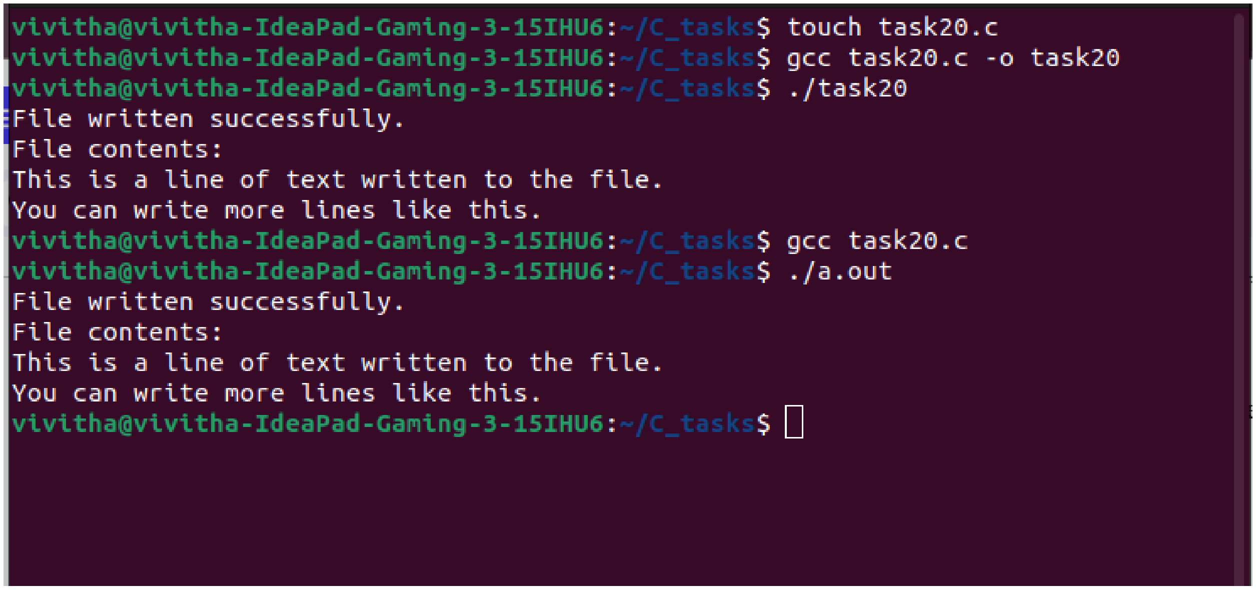
20.Write a C program to make , write and read file.

Code:

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

int main() {
   FILE *file;

   file = fopen("example.txt", "w");

   if (file == NULL) {
      printf("Failed to open the file for writing.\n");
      return 1;
   }

   fprintf(file, "This is a line of text written to the file.\n");
   fprintf(file, "You can write more lines like this.\n");

   fclose(file);

   printf("File written successfully.\n");
   file = fopen("example.txt", "r");

   if (file == NULL) {
      printf("Failed to open the file for reading.\n");
      return 1;
   }
```

```
    char buffer[100];

    printf("File contents:\n");
    while (fgets(buffer, sizeof(buffer), file) != NULL) {
        printf("%s", buffer);
    }

    fclose(file);

    return 0;
}
```

Output:



```
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ touch task20.c
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task20.c -o task20
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./task20
File written successfully.
File contents:
This is a line of text written to the file.
You can write more lines like this.
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ gcc task20.c
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$ ./a.out
File written successfully.
File contents:
This is a line of text written to the file.
You can write more lines like this.
vivitha@vivitha-IdeaPad-Gaming-3-15IHU6:~/C_tasks$
```