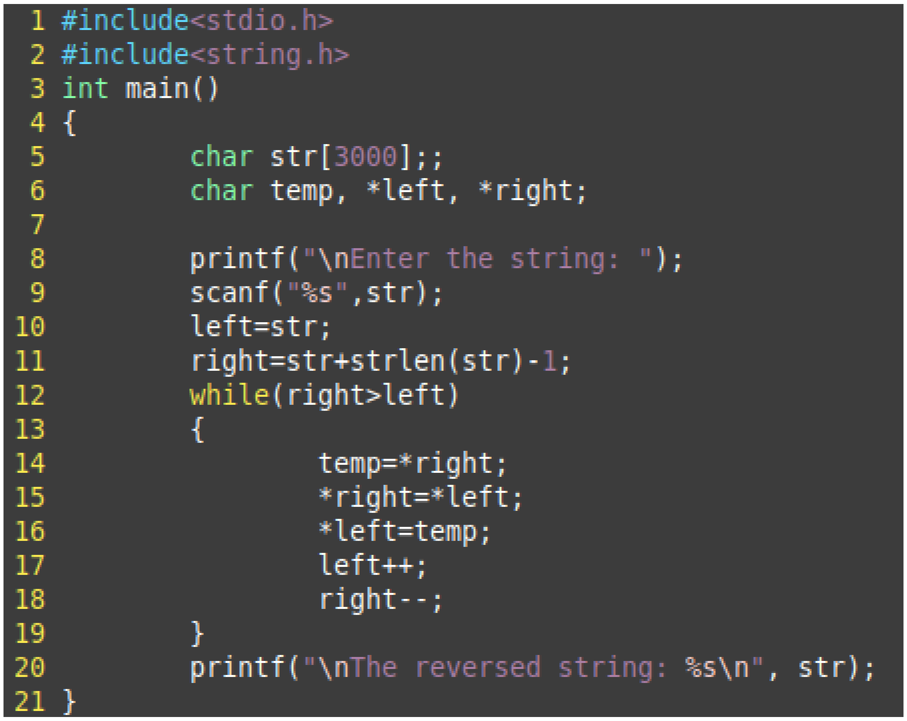
Birla Institute of Technology & Science, Pilani, Hyderabad Campus

First Semester 2020-2021
Computer Programming [CS F111] Lab 12

Practice Programs (Complete atleast any 2 towards submission):

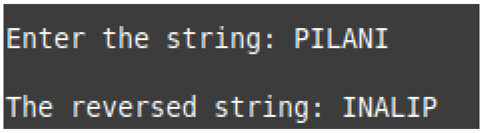
1. Write a program to reverse the characters of a String using pointers.

Code:



```c
#include<stdio.h>
#include<string.h>
int main()
{
        char str[3000];;
        char temp, *left, *right;

        printf("\nEnter the string: ");
        scanf("%s",str);
        left=str;
        right=str+strlen(str)-1;
        while(right>left)
        {
                temp=*right;
                *right=*left;
                *left=temp;
                left++;
                right--;
        }
        printf("\nThe reversed string: %s\n", str);
}
```

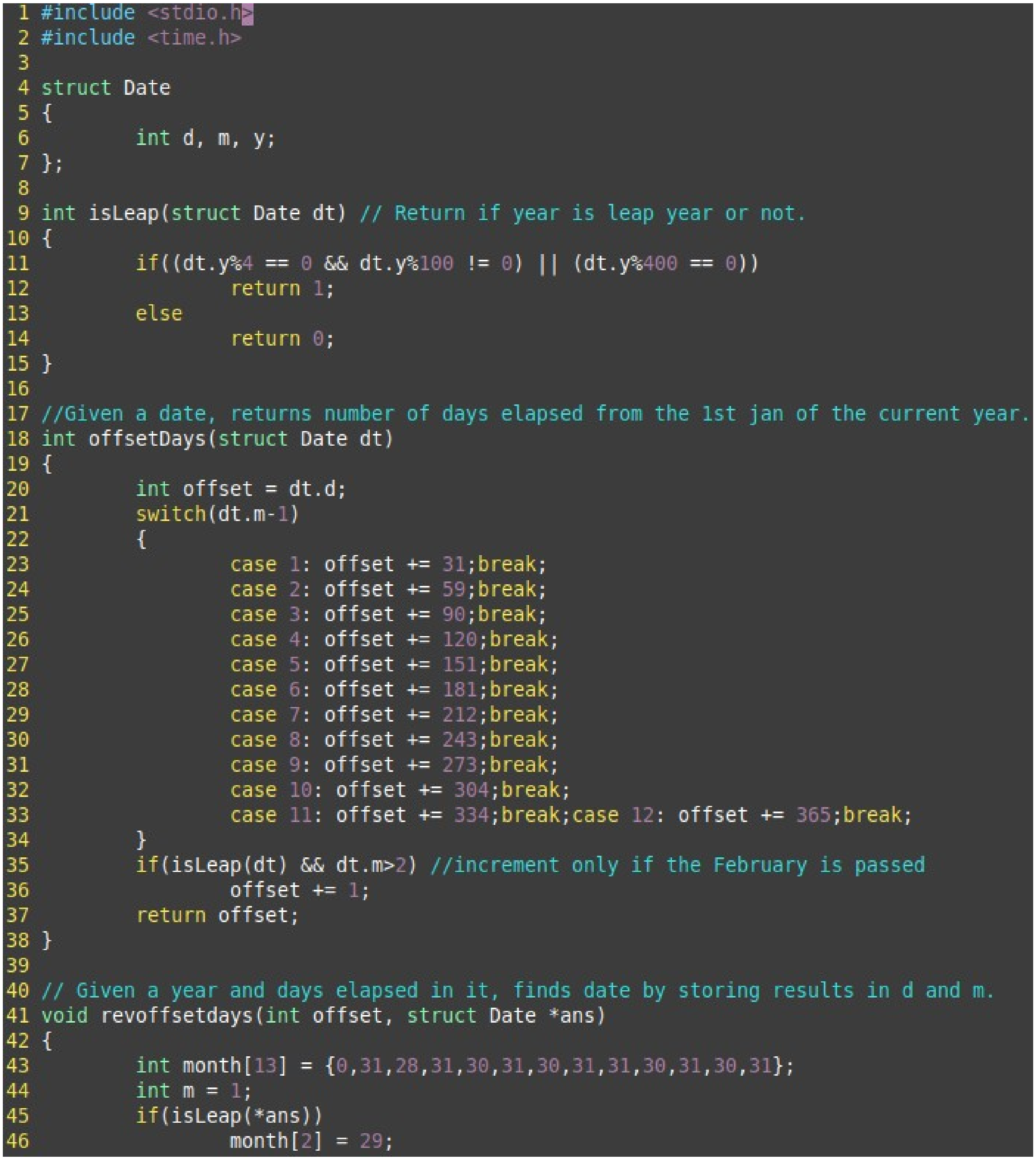
Sample Input/Output:



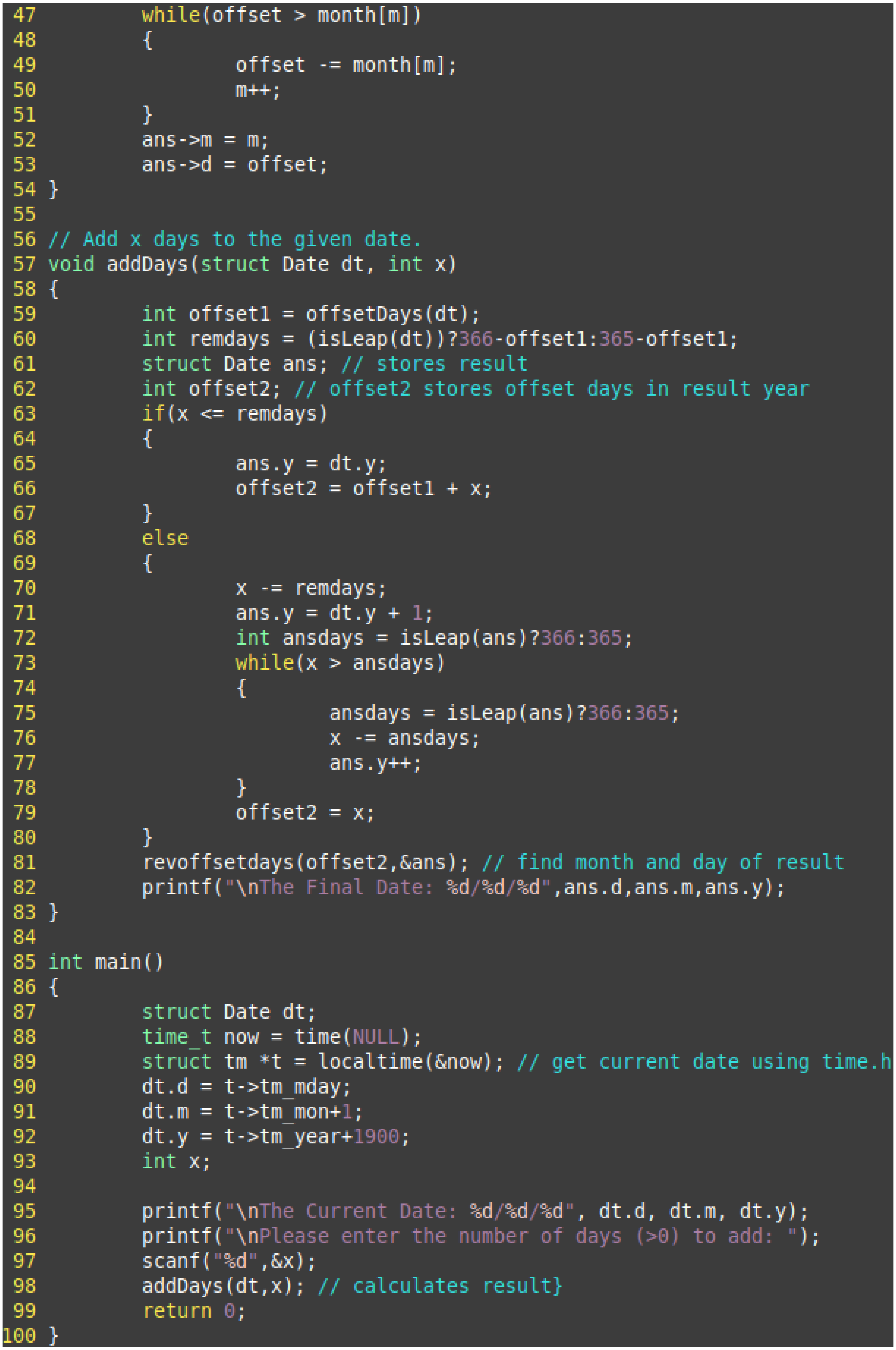
```
Enter the string: PILANI

The reversed string: INALIP
```

2. Write a C program to create a structure named Date. The structure should have day, month and year as inputs. Store the current date in the structure. Take inputs ‘x’ from the user where x is the number of days. Add x to current date and display the final date.

Code:

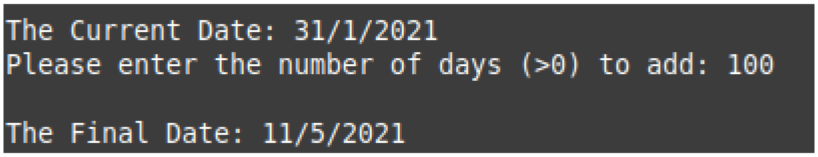


```c
#include <stdio.h>
#include <time.h>

struct Date
{
        int d, m, y;
};

int isLeap(struct Date dt) // Return if year is leap year or not.
{
        if((dt.y%4 == 0 && dt.y%100 != 0) || (dt.y%400 == 0))
                return 1;
        else
                return 0;
}

//Given a date, returns number of days elapsed from the 1st jan of the current year.
int offsetDays(struct Date dt)
{
        int offset = dt.d;
        switch(dt.m-1)
        {
                case 1: offset += 31;break;
                case 2: offset += 59;break;
                case 3: offset += 90;break;
                case 4: offset += 120;break;
                case 5: offset += 151;break;
                case 6: offset += 181;break;
                case 7: offset += 212;break;
                case 8: offset += 243;break;
                case 9: offset += 273;break;
                case 10: offset += 304;break;
                case 11: offset += 334;break;case 12: offset += 365;break;
        }
        if(isLeap(dt) && dt.m>2) //increment only if the February is passed
                offset += 1;
        return offset;
}

// Given a year and days elapsed in it, finds date by storing results in d and m.
void revoffsetdays(int offset, struct Date *ans)
{
        int month[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
        int m = 1;
        if(isLeap(*ans))
                month[2] = 29;
```



```c
        while(offset > month[m])
        {
                offset -= month[m];
                m++;
        }
        ans->m = m;
        ans->d = offset;
}

// Add x days to the given date.
void addDays(struct Date dt, int x)
{
        int offset1 = offsetDays(dt);
        int remdays = (isLeap(dt))?366-offset1:365-offset1;
        struct Date ans; // stores result
        int offset2; // offset2 stores offset days in result year
        if(x <= remdays)
        {
                ans.y = dt.y;
                offset2 = offset1 + x;
        }
        else
        {
                x -= remdays;
                ans.y = dt.y + 1;
                int ansdays = isLeap(ans)?366:365;
                while(x > ansdays)
                {
                        ansdays = isLeap(ans)?366:365;
                        x -= ansdays;
                        ans.y++;
                }
                offset2 = x;
        }
        revoffsetdays(offset2,&ans); // find month and day of result
        printf("\nThe Final Date: %d/%d/%d",ans.d,ans.m,ans.y);
}

int main()
{
        struct Date dt;
        time_t now = time(NULL);
        struct tm *t = localtime(&now); // get current date using time.h
        dt.d = t->tm_mday;
        dt.m = t->tm_mon+1;
        dt.y = t->tm_year+1900;
        int x;

        printf("\nThe Current Date: %d/%d/%d", dt.d, dt.m, dt.y);
        printf("\nPlease enter the number of days (>0) to add: ");
        scanf("%d",&x);
        addDays(dt,x); // calculates result}
        return 0;
}
```

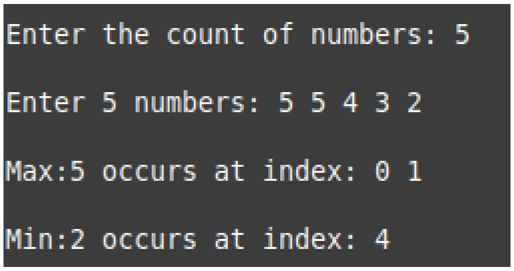
Sample Input/Output:



```
The Current Date: 31/1/2021
Please enter the number of days (>0) to add: 100

The Final Date: 11/5/2021
```

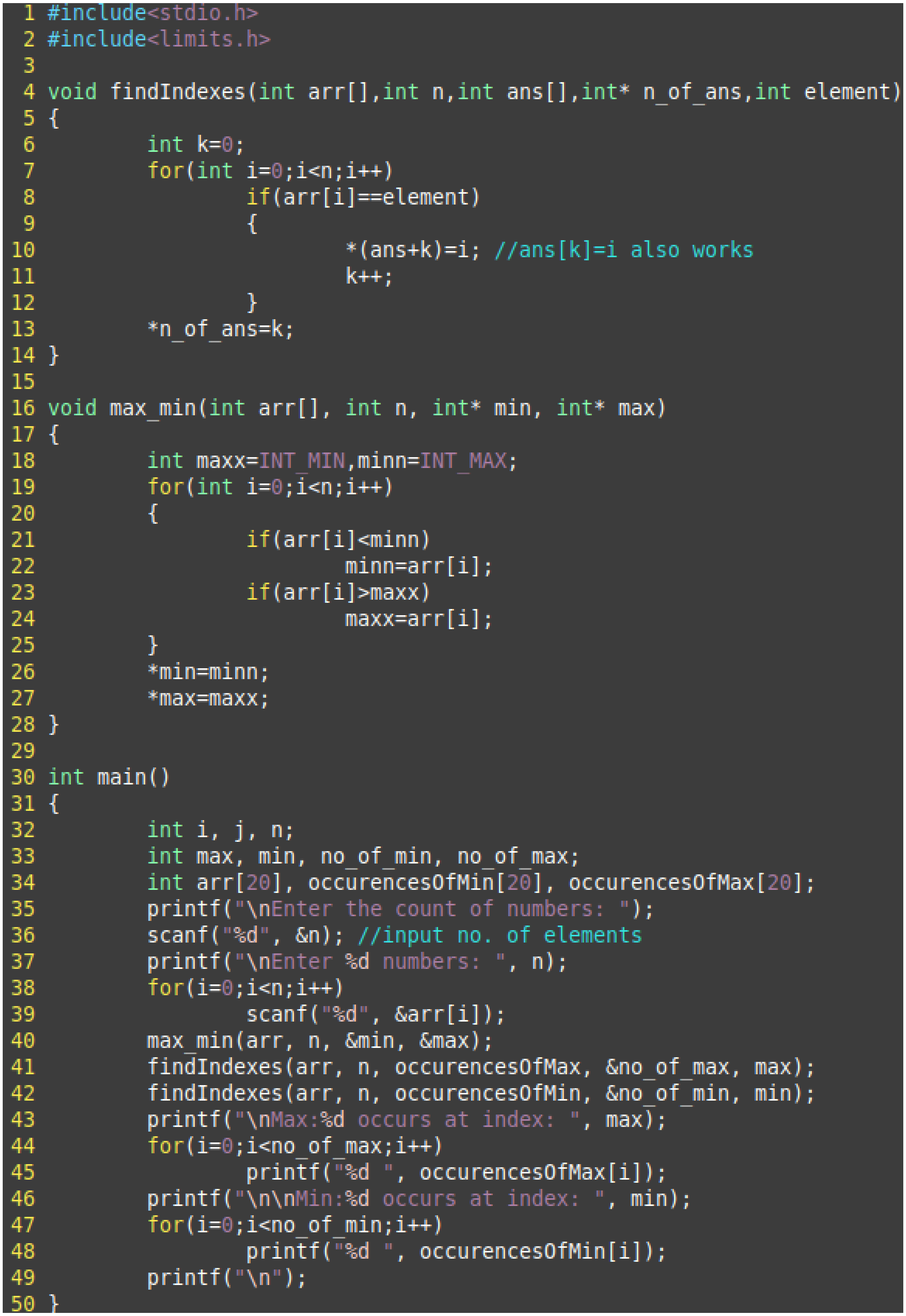
3. Write a C program to pass multiple value to a function using pointers. Find the minimum and maximum values in a given array using the functions. Print the indices of occurrences of min and max.

Sample Input/Output:



```
Enter the count of numbers: 5

Enter 5 numbers: 5 5 4 3 2

Max:5 occurs at index: 0 1

Min:2 occurs at index: 4
```

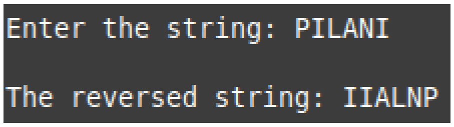
Code:



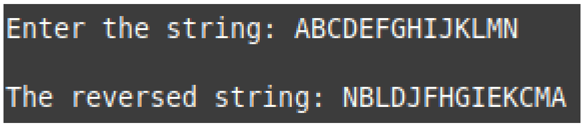
```
#include<stdio.h>
#include<limits.h>

void findIndexes(int arr[],int n,int ans[],int* n_of_ans,int element)
{
        int k=0;
        for(int i=0;i<n;i++)
                if(arr[i]==element)
                {
                        *(ans+k)=i; //ans[k]=i also works
                        k++;
                }
        *n_of_ans=k;
}

void max_min(int arr[], int n, int* min, int* max)
{
        int maxx=INT_MIN,minn=INT_MAX;
        for(int i=0;i<n;i++)
        {
                if(arr[i]<minn)
                        minn=arr[i];
                if(arr[i]>maxx)
                        maxx=arr[i];
        }
        *min=minn;
        *max=maxx;
}

int main()
{
        int i, j, n;
        int max, min, no_of_min, no_of_max;
        int arr[20], occurencesOfMin[20], occurencesOfMax[20];
        printf("\nEnter the count of numbers: ");
        scanf("%d", &n); //input no. of elements
        printf("\nEnter %d numbers: ", n);
        for(i=0;i<n;i++)
                scanf("%d", &arr[i]);
        max_min(arr, n, &min, &max);
        findIndexes(arr, n, occurencesOfMax, &no_of_max, max);
        findIndexes(arr, n, occurencesOfMin, &no_of_min, min);
        printf("\nMax:%d occurs at index: ", max);
        for(i=0;i<no_of_max;i++)
                printf("%d ", occurencesOfMax[i]);
        printf("\n\nMin:%d occurs at index: ", min);
        for(i=0;i<no_of_min;i++)
                printf("%d ", occurencesOfMin[i]);
        printf("\n");
}
```

Exercise Problems (Complete atleast any 3 towards submission):

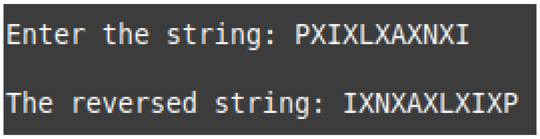
1. Modify the practice program-1 to reverse alternate characters of a string using pointers. You cannot introduce any other additional variable.
   Sample Input/Outputs:



```
Enter the string: PILANI

The reversed string: IIALNP
```



```
Enter the string: ABCDEFGHIJKLMN

The reversed string: NBLDJFHGIEKCMA
```



```
Enter the string: PXIXLXAXNXI

The reversed string: IXNXAXLXIXP
```

2. Modify the practice program-2 to read any date and the ‘x’ (number of days) to print the final date.
3. Modify the practice program-3 to additionally print the indices at which median of the given numbers occur.
4. Write a C program to add two complex numbers by passing structure to a function.

*************ALL THE BEST*************

NOTE: Upload the screenshots of the Practice programs and Exercise programs along with the displayed results into your corresponding Google Classroom.