

Birla Institute of Technology & Science, Pilani, Hyderabad Campus

Second Semester 2020-2021

CS F111: Computer Programming (Lab 12)

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

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char str[3000];
6     char temp, *left, *right;
7
8     printf("\nEnter the string: ");
9     scanf("%s",str);
10    left=str;
11    right=str+strlen(str)-1;
12    while(right>left)
13    {
14        temp=*right;
15        *right=*left;
16        *left=temp;
17        left++;
18        right--;
19    }
20    printf("\nThe reversed string: %s\n", str);
21 }
```

Sample Input/Output:

```
Enter the string: PILANI
The reversed string: INALIP
```

Q.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.

```
1 #include <stdio.h>
2 #include <time.h>
3
4 struct Date
5 {
6     int d, m, y;
7 };
8
9 int isLeap(struct Date dt) // Return if year is leap year or not.
10 {
11     if((dt.y%4 == 0 && dt.y%100 != 0) || (dt.y%400 == 0))
12         return 1;
13     else
14         return 0;
15 }
16
17 //Given a date, returns number of days elapsed from the 1st jan of the current year.
18 int offsetDays(struct Date dt)
19 {
20     int offset = dt.d;
21     switch(dt.m-1)
22     {
23         case 1: offset += 31;break;
24         case 2: offset += 59;break;
25         case 3: offset += 90;break;
26         case 4: offset += 120;break;
27         case 5: offset += 151;break;
28         case 6: offset += 181;break;
29         case 7: offset += 212;break;
30         case 8: offset += 243;break;
31         case 9: offset += 273;break;
32         case 10: offset += 304;break;
33         case 11: offset += 334;break;case 12: offset += 365;break;
34     }
35     if(isLeap(dt) && dt.m>2) //increment only if the February is passed
36         offset += 1;
37     return offset;
38 }
39
40 // Given a year and days elapsed in it, finds date by storing results in d and m.
41 void revoffsetdays(int offset, struct Date *ans)
42 {
43     int month[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
44     int m = 1;
45     if(isLeap(*ans))
46         month[2] = 29;
```

```

47     while(offset > month[m])
48     {
49         offset -= month[m];
50         m++;
51     }
52     ans->m = m;
53     ans->d = offset;
54 }
55
56 // Add x days to the given date.
57 void addDays(struct Date dt, int x)
58 {
59     int offset1 = offsetDays(dt);
60     int remdays = (isLeap(dt))?366-offset1:365-offset1;
61     struct Date ans; // stores result
62     int offset2; // offset2 stores offset days in result year
63     if(x <= remdays)
64     {
65         ans.y = dt.y;
66         offset2 = offset1 + x;
67     }
68     else
69     {
70         x -= remdays;
71         ans.y = dt.y + 1;
72         int ansdays = isLeap(ans)?366:365;
73         while(x > ansdays)
74         {
75             ansdays = isLeap(ans)?366:365;
76             x -= ansdays;
77             ans.y++;
78         }
79         offset2 = x;
80     }
81     revoffsetdays(offset2,&ans); // find month and day of result
82     printf("\nThe Final Date: %d/%d/%d",ans.d,ans.m,ans.y);
83 }
84
85 int main()
86 {
87     struct Date dt;
88     time_t now = time(NULL);
89     struct tm *t = localtime(&now); // get current date using time.h
90     dt.d = t->tm_mday;
91     dt.m = t->tm_mon+1;
92     dt.y = t->tm_year+1900;
93     int x;
94
95     printf("\nThe Current Date: %d/%d/%d", dt.d, dt.m, dt.y);
96     printf("\nPlease enter the number of days (>0) to add: ");
97     scanf("%d",&x);
98     addDays(dt,x); // calculates result
99     return 0;
100 }

```

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
```

Q.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.

```
1 #include<stdio.h>
2 #include<limits.h>
3
4 void findIndexes(int arr[],int n,int ans[],int* n_of_ans,int element)
5 {
6     int k=0;
7     for(int i=0;i<n;i++)
8         if(arr[i]==element)
9         {
10             *(ans+k)=i; //ans[k]=i also works
11             k++;
12         }
13     *n_of_ans=k;
14 }
15
16 void max_min(int arr[], int n, int* min, int* max)
17 {
18     int maxx=INT_MIN,minn=INT_MAX;
19     for(int i=0;i<n;i++)
20     {
21         if(arr[i]<minn)
22             minn=arr[i];
23         if(arr[i]>maxx)
24             maxx=arr[i];
25     }
26     *min=minn;
27     *max=maxx;
28 }
29
30 int main()
31 {
32     int i, j, n;
33     int max, min, no_of_min, no_of_max;
34     int arr[20], occurrencesOfMin[20], occurrencesOfMax[20];
35     printf("\nEnter the count of numbers: ");
36     scanf("%d", &n); //input no. of elements
37     printf("\nEnter %d numbers: ", n);
38     for(i=0;i<n;i++)
39         scanf("%d", &arr[i]);
40     max_min(arr, n, &min, &max);
41     findIndexes(arr, n, occurrencesOfMax, &no_of_max, max);
42     findIndexes(arr, n, occurrencesOfMin, &no_of_min, min);
43     printf("\nMax:%d occurs at index: ", max);
44     for(i=0;i<no_of_max;i++)
45         printf("%d ", occurrencesOfMax[i]);
46     printf("\n\nMin:%d occurs at index: ", min);
47     for(i=0;i<no_of_min;i++)
48         printf("%d ", occurrencesOfMin[i]);
49     printf("\n");
50 }
```

```
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
```

Exercise Problem:

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
```