PRACTICAL NO : 01 (A-2)

Title : Write a Python program to store marks scored in subject “Fundamental of Data Structure” by N students in the class. Write functions to compute following:

a) The average score of class

b) Highest score and lowest score of class

c) Count of students who were absent for the test

d) Display mark with highest frequency

Main program :

```
def avg(marks):
 add=0
 count=0
 for i in range(len(marks)):
    if marks[i] !=-1:
    add+=marks[i]
    count+=1
 avg=add/count
 print('Total Marks:',add)
 print('Average Marks ',avg)

def max(marks):
 maxi=0
 count=0
 for i in range(len(marks)):
    if maxi<marks[i]:
        maxi=marks[i]
 return maxi

def min(marks):
 mini=0
 count=0
 for i in range(len(marks)):
    if marks[i] != -1:
        mini=marks[i]
    break
 for j in range(i+1,len(marks)):
    if marks[j] !=-1 and marks[j]<mini:
        mini=marks[j]
 return mini

def absentCnt(marks):
 count=0
 for i in range(len(marks)):
    if marks[i] ==-1:
        count +=1
 return (count)

def maxfreq(marks):
 i=0
 max=0
 print('Marks---- > frequency count')
 for obj in marks:
    if marks.index(obj) == i:
        print(obj,'--- >',marks.count(obj))
    if marks.count(obj)>max:
        max=marks.count(obj)
 mark=obj
 i+=1
 return (marks,max)

n=int(input('Enter total number of students:'))
marks=[]
for i in range(n):
 mks=int(input('Enter marks of Student'+str(i+1)+ ' : '))
 marks.append(mks)

while(1):
        print('/******** Main Menu*******/')
        print('1.Average \n2.Max Marks \n3.Min Marks \n4.Absent Students\n5.Highest Frequency \n6.Exit')
        ch=int(input('Enter your choice:'))
        if ch==1:
            (avg(marks))
        elif ch==2:
            print("Maximum Marks are:",max(marks))
        elif ch==3:
            print("Minimum Marks are:",min(marks))
        elif ch==4:
            print("Absent is:",absentCnt(marks))
        elif ch==5:
            marks,count=maxfreq(marks)
            print("Highest Frequency:",maxfreq(marks))
        elif ch==6:
            break
```

OUTPUT :

```
Enter total number of students:5
Enter marks of Student1 : 56
Enter marks of Student2 : 89
Enter marks of Student3 : 87
Enter marks of Student4 : 57
Enter marks of Student5 : 36
/******** Main Menu*******/
```

```
1.Average
2.Max Marks
3.Min Marks
4.Absent Students
5.Highest Frequency
6.Exit
Enter your choice:1
Total Marks: 325
Average Marks  65.0
/******** Main Menu*******/
1.Average
2.Max Marks
3.Min Marks
4.Absent Students
5.Highest Frequency
6.Exit
Enter your choice:2
Maximum Marks are: 89
/******** Main Menu*******/
1.Average
2.Max Marks
3.Min Marks
4.Absent Students
5.Highest Frequency
6.Exit
Enter your choice:3
Minimum Marks are: 36
/******** Main Menu*******/
1.Average
2.Max Marks
3.Min Marks
4.Absent Students
5.Highest Frequency
6.Exit
Enter your choice:4
Absent is: 0
/******** Main Menu*******/
1.Average
2.Max Marks
3.Min Marks
4.Absent Students
```

```
5.Highest Frequency
6.Exit
Enter your choice:5
Marks---- > frequency count
56 ----> 1
Marks---- > frequency count
56 ----> 1
Highest Frequency: ([56, 89, 87, 57, 36], 1)
/******** Main Menu*******/
1.Average
2.Max Marks
3.Min Marks
4.Absent Students
5.Highest Frequency
6.Exit
Enter your choice:6
```

PRACTICAL NO : 02 (A-5)

TITLE : Write a Python program to compute following operations on String: a) To display word with the longest length b) To determines the frequency of occurrence of particular character in the string c) To check whether given string is palindrome or not d) To display index of first appearanceof the substring e) To count the occurrences of each word in a given string.

Main program :

```
def longest():
    str1=input("Enter the string :")
    str=str1+"  "
    longest_word= " "
    max_length=0
    current_word=" "
    current_length=0

    for word in str:
        if word == ''or word == str[-1]:

            if current_length > max_length:
                max_length = current_length
                longest_word = current_word

            current_length = ""
            current_length = 0
        else:
            current_word += word
            current_length += 1
            print("The word with longest length is :",longest_word)

def occur():
    str = input("Enter the string :")
    ch = input("Which character do you want to search in string :")
    cnt=0

     for char in str:
         if char==ch:
            cnt+=1

    print(ch,"Occured",cnt,"times")

def palindrome():
    str=input("Enter String To Check Palindrome :")
    str1=str[::-1]

    if str1==str:
        print("It is a Palindrome")
    else:
        print("It is not a Palindrome")

def find_sub():
    str = input("Enter the string : ")
    sub_str = input("Enter the substring to search for :")
    index = str.find(sub_str)

    if index != -1:
        print(f"The index of the occurence of '{sub_str}'")
    else:
        print(f"'{sub_str}'not found in the substring")

def occur_word():
    str1 = input("Enter the string :")
    str = str1+" "
    word_list ={}
    current_word=""

    for word in str:
         if word == ' ' or word ==str[-1]:
             if current_word in word_list:
                word_list[current_word]+=1
            else:
                word_list[current_word] = 1
            current_word = ""

        else:
            current_word+= word
    print("Word--> Occurance")
    for key in word_list:
        print(f'{key} -->{word_list[key]}')

#main code
while(1):
    print("\n\t****STRING OPERATION****\n")
    print("1.Word with longest length\n2.Frequency of a character\n3.Check Palindrome\n4.Index of first
substring\n5.Count occurance of each word in string\n6.Exit\n")
    ch = int(input("Enter Your Choice :"))
    print("")

    if ch == 1:
        longest()

    elif ch == 2:
        occur()

    elif ch == 3:
        palindrome()

    elif ch == 4:
```

```
    find_sub()

  elif ch == 5:
    occur_word()

  elif ch ==6:
    print("\t\t***TERMINATED SUCCESSFULLY***\n")
    break

  else:
    Exit
```

Output:-

```
  ****STRING OPERATION****

1.Word with longest length
2.Frequency of a character
3.Check Palindrome
4.Index of first substring
5.Count occurance of each word in string
6.Exit

Enter Your Choice :1

Enter the string :vinit
The word with longest length is :
The word with longest length is :
The word with longest length is :
The word with longest length is :
The word with longest length is :

        ****STRING OPERATION****

1.Word with longest length
2.Frequency of a character
3.Check Palindrome
4.Index of first substring
5.Count occurance of each word in string
6.Exit
```

```
Enter Your Choice :2

Enter the string :vinit
Which character do you want to search in string :i
i Occured 2 times

        ****STRING OPERATION****

1.Word with longest length
2.Frequency of a character
3.Check Palindrome
4.Index of first substring
5.Count occurance of each word in string
6.Exit

Enter Your Choice :3

Enter String To Check Palindrome :nitin
It is a Palindrome

        ****STRING OPERATION****

1.Word with longest length
2.Frequency of a character
3.Check Palindrome
4.Index of first substring
5.Count occurance of each word in string
6.Exit

Enter Your Choice :4

Enter the string : king
Enter the substring to search for :i
The index of the occurence of 'i'
```

```
        ****STRING OPERATION****

1.Word with longest length
2.Frequency of a character
3.Check Palindrome
4.Index of first substring
5.Count occurance of each word in string
6.Exit

Enter Your Choice :5

Enter the string :Enginnering
Word--> Occurance
Enginnering -->1

        ****STRING OPERATION****

1.Word with longest length
2.Frequency of a character
3.Check Palindrome
4.Index of first substring
5.Count occurance of each word in string
6.Exit

Enter Your Choice :6

                ***TERMINATED SUCCESSFULLY***
```

PRACTICAL NO : 03 (A-8)

TITLE : Write a Python program that determines the location of a saddle point of matrix if one exists. An m x n matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row i and the largest value in j.

Main program :

```
def take_matrix(m, n):
   out = []
   for i in range(m):
      row = []
      for j in range(n):
         element = int(input(f"Enter Element [{i}][{j}] :"))
         row.append(element)
      out.append(row)
   return out


def saddle():
             #take no. of rows and columns
           m=int(input("Enter Rows Of Matrix : "))
           n=int(input("Enter Columns Of Matrix : "))
           #take matrices A and B
           print("\n\tMATRIX A")
           A = take_matrix(m,n)

           for row in range(m):
                      min_row = A[row][0]
                      col_index = 0

                      for col in range(n):
                                 if (min_row > A[row][col]):
                                            min_row = A[row][col]
```

```
                col_index = col

    k = 0
    for k in range(n):

        if (min_row < A[k][col_index]):
            break
        k += 1
    if k == n:
        print("Value of Saddle Point ",min_row)
        print("Index of ", min_row ,"is ",row," ",col_index)
        return True

#main

saddle()
```

Output:-

```
Enter  Rows  Of  Matrix  :  2
Enter Columns Of Matrix : 2

   MATRIX A
Enter Element [0][0] :5
Enter Element [0][1] :6
Enter Element [1][0] :8
Enter Element [1][1] :9
Value of Saddle Point 8
Index of 8 is  1  0
```

PRACTICAL NO : 04 (A-9)

TITLE: - Write a Python program to compute following computation on matrix:

a) Addition of two matrices

b) Subtraction of two matrices

c) Multiplication of two matrices

d) Transpose of a matrix

Main program :

```
row=int(input("Give the no. of row:"))
column=int(input("Give the no. of col:"))
matrix1=[[0 for i in range(column)] for i in range (row)]
ans=[[0,0],[0,0],[0,0]]
for i in range(row):
        for j in range(column):
                        matrix1[i][j]=int(input(f"Enter value for {i}{j}:"))


print(matrix1)
print("display Matrix")
for i in range(row):
        for j in range(column):
                print(matrix1[i][j],end=" ")
        print()



row=int(input("Give the no. of row:"))
column=int(input("Give the no. of col:"))
matrix2=[[0 for i in range(column)] for i in range (row)]

for i in range(row):
        for j in range(column):
            matrix2[i][j]=int(input(f"Enter value for {i}{j}:"))


print(matrix2)
print("display Matrix")
for i in range(row):
    for j in range(column):
        print(matrix2[i][j],end=" ")
    print()



#addition
for i in range(row):
    for j in range(column):
        ans[i][j]=matrix1[i][j] + matrix2[i][j]
print("Addition of 2 matrix is:",ans)

#substration
for i in range(row):
    for j in range(column):
        ans[i][j]=matrix1[i][j] - matrix2[i][j]
print("substration of 2 matrix is:",ans)

#multiplication
for i in range(row):
    for i in range(column):
        ans[i][j]=matrix1[i][j]*matrix2[i][j]
print("Multiplication of 2 matrix is:",ans)

#Transpose
def transpose(m,r,c):
    ans=[[0 for i in range(r)] for j in range(c)]

    for i in range(r):
        for j in range(c):
            ans[i][j]=m[j][i]
    return ans
print("Transpose:",transpose(matrix1,row,column))
```

output:-

```
Give the no. of row:2
Give the no. of col:2
Enter value for 00:4
Enter value for 01:5
Enter value for 10:2
Enter value for 11:3
[[4, 5], [2, 3]]
display Matrix
4 5
2 3
Give the no. of row:2
Give the no. of col:2
Enter value for 00:5
Enter value for 01:4
Enter value for 10:3
Enter value for 11:2
[[5, 4], [3, 2]]
display Matrix
5 4
3 2
Addition of 2 matrix is: [[9, 9], [5, 5], [0, 0]]
substration of 2 matrix is: [[-1, 1], [-1, 1], [0, 0]]
Multiplication of 2 matrix is: [[-1, 20], [-1, 6], [0, 0]]
Transpose: [[4, 2], [5, 3]]
```

PRACTICAL NO : 05 (B-14)

TITLE : Write a Python program to store first year percentage of students in array.

Writefunction for sorting array of floating point numbers in ascending orderusing

a) **Selection Sort**

b) **Bubble sort and display top five scores**

Main program :

```
def bubble(percent):
    for i in range(len(percent)-1):
        for  j in range(len(percent)-i-1):
            if percent[j]>percent[j+1]:
                percent[j],percent[j+1]=percent[j+1],percent[j]

    return percent

def selectionSort(array, size):

    for ind in range(size):
        min_index = ind

        for j in range(ind + 1, size):

            if array[j] < array[min_index]:
                min_index = j

        (array[ind], array[min_index]) = (array[min_index], array[ind])

    return array

def display(percent):
    for i in range(len(percent)-1,4,-1):
        print(percent[i])

#main code
```

```
#take user input and store in list
n = int(input("Enter No. of students : "))
percent = []
for i in range(n):
    mks=float(input("Enter percentage of student "+str(i+1)+" : "))
    percent.append(mks)

while(1):
    print("\n\t***** Sorting Menu *****")
    ch=int(input("1. Selection Sort, 2.Bubble Sort and Top 5 Scores, 3.Exit "))

    if ch==1:
        print("\nResult of Selection Sort is : ",selectionSort(percent,n))
    elif ch==2:
        print("\nResult of Bubble Sort is : ",bubble(percent))
        print("Top 5 Scores are : ")
        display(percent)
    elif ch==3:
        print("\t***** TERMINATED SUCCESSFULLY *****")
        break
    else:
        print("Invalid Choice")
```

Output:-

```
Enter No. of students : 8
Enter percentage of student 1 : 65
Enter percentage of student 2 : 98
Enter percentage of student 3 : 56
Enter percentage of student 4 : 35
Enter percentage of student 5 : 78
Enter percentage of student 6 : 96
Enter percentage of student 7 : 97
Enter percentage of student 8 : 82
***** Sorting Menu *****
1. Selection Sort, 2.Bubble Sort and Top 5 Scores, 3.Exit 1
Result of Selection Sort is :  [35.0, 56.0, 65.0, 78.0, 82.0, 96.0, 97.0, 98.0]

        ***** Sorting Menu *****
1. Selection Sort, 2.Bubble Sort and Top 5 Scores, 3.Exit 2
Result of Bubble Sort is :  [35.0, 56.0, 65.0, 78.0, 82.0, 96.0, 97.0, 98.0]
```

```
Top 5 Scores are :
98.0
97.0
96.0

        ***** Sorting Menu *****
1. Selection Sort, 2.Bubble Sort and Top 5 Scores, 3.Exit 3
***** TERMINATED SUCCESSFULLY *****
```

PRACTICAL NO : 05 (B-14)

TITLE:- Write a Python program to store first year percentage of students in array.

Writefunction for sorting array of floating point numbers in ascending order using

a) Insertion Sort

b) Shell sort and display top five scores

Main program :

```
def insertionsort(arr):

        for i in range(1, len(arr)):

                key = arr[i]

                j = i-1
                while j >= 0 and key < arr[j] :
                                arr[j + 1] = arr[j]
                                j -= 1
                arr[j + 1] = key
        return arr

def shellsort(array, n):

    # Rearrange elements at each n/2, n/4, n/8, ... intervals
    interval = n // 2
    while interval > 0:
        for i in range(interval, n):
            temp = array[i]
            j = i
            while j >= interval and array[j - interval] > temp:
                array[j] = array[j - interval]
                j -= interval

            array[j] = temp
        interval //= 2
    return array

def display(percent):
        for i in range(len(percent)-1,4,-1):
                print(percent[i])

#main code

#take user input and store in list
n = int(input("Enter No. of students : "))
percent = []
for i in range(n):
        mks=float(input("Enter percentage of student "+str(i+1)+" : "))
        percent.append(mks)

while(1):
        print("\n\t***** Sorting Menu *****")
        ch=int(input("1. Insertion Sort, 2.Shell Sort and Top 5 Scores, 3.Exit "))

        if ch==1:
                print("\nResult of Insertion Sort is : ",insertionsort(percent))
        elif ch==2:
                print("\nResult of Shell Sort is : ",shellsort(percent,n))
                print("Top 5 Scores are : ")
                display(percent)
        elif ch==3:
                print("\n\t***** TERMINATED SUCCESSFULLY *****")
                break
        else:
                print("Invalid Choice")
```

Output:-

```
Enter No. of students : 7
Enter percentage of student 1 : 58
Enter percentage of student 2 : 96
Enter percentage of student 3 : 78
Enter percentage of student 4 : 98
Enter percentage of student 5 : 56
Enter percentage of student 6 : 23
Enter percentage of student 7 : 15
***** Sorting Menu *****
1. Insertion Sort, 2.Shell Sort and Top 5 Scores, 3.Exit 1
Result of Insertion Sort is :  [15.0, 23.0, 56.0, 58.0, 78.0, 96.0, 98.0]

        ***** Sorting Menu *****
1. Insertion Sort, 2.Shell Sort and Top 5 Scores, 3.Exit 2
Result of Shell Sort is :  [15.0, 23.0, 56.0, 58.0, 78.0, 96.0, 98.0]
Top 5 Scores are :
98.0
96.0

        ***** Sorting Menu *****
1. Insertion Sort, 2.Shell Sort and Top 5 Scores, 3.Exit 3
***** TERMINATED SUCCESSFULLY *****
```

****

PRACTICAL NO : 07(B-16)

TITLE : : Write a Python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.

Main Program

```
def partition(array, low, high):

    pivot = array[high]
    i = low - 1

    for j in range(low, high):

        if array[j] <= pivot:
            i = i + 1
            (array[i], array[j]) = (array[j], array[i])

    (array[i + 1], array[high]) = (array[high], array[i + 1])
    return i + 1

def quickSort(array, low, high):
    if low < high:
        pi = partition(array, low, high)

        quickSort(array, low, pi - 1)
        quickSort(array, pi + 1, high)

def display(percent):
    for i in range(n-1,n-6,-1):
        print(percent[i])

#main code
print("\t *****Perform Quick Sort*****")
n = int(input("Enter No. of students : "))
percent = []
for i in range(n):
    mks=float(input("Enter percentage of student "+str(i+1)+" : "))
    percent.append(mks)

size = len(percent)
quickSort(percent, 0, size-1)
while(1):
    print("\n1.Quick Sort \n2.Top 5 scores are: \n3.Exit")
    ch=int(input("Enter your choice:"))
    if ch==1:
        print("Sorted Array using quick sort is :", percent)
    if ch==2:
        print("Top 5 scores are : ",display(percent))
    if ch==3:
        break
```

***********************************OUTPUT******************************

```
*****Perform Quick Sort*****
Enter No. of students : 10
Enter percentage of student 1 : 98
Enter percentage of student 2 : 85
Enter percentage of student 3 : 47
Enter percentage of student 4 : 72
Enter percentage of student 5 : 63
Enter percentage of student 6 : 58
Enter percentage of student 7 : 71
Enter percentage of student 8 : 90
Enter percentage of student 9 : 64
Enter percentage of student 10 : 77
1.Quick Sort
2. Top 5 scores are:
3. Exit
Enter your choice:1
Sorted Array using quick sort is : [47.0, 58.0, 63.0, 64.0, 71.0, 72.0, 77.0, 85.0, 90.0, 98.0]

1.Quick Sort
2.Top 5 scores are:
3.Exit
Enter your choice:2
98.0
90.0
85.0
77.0
72.0
Top 5 scores are :  None

1.Quick Sort
2.Top 5 scores are:
3.Exit
Enter your choice:3
>
```

PRACTICAL NO : 08 (C-19)

TITLE :- Department of Computer Engineering has student's club named 'Pinnacle Club'.

Students of second, third and final year of department can be granted membership on request. Similarly onemay cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club member‘s information using singly linked list. Store student PRN and Name. Write functions to:

a) Add and delete the members as well as president or even secretary.

b) Compute total number of members of club

c) Display members

d) Two linked lists exists for two divisions. Concatenate two lists.

Main program : -

```
#include<iostream>
#include<string.h>
using namespace std;
struct node
{    int  prn,rollno;
      char name[50];
      struct node *next;
};
class info
{        node
*s=NULL,*head1=NULL,*temp1=NULL,*head2=NULL,*temp2=NULL,*head=NULL,*temp=NULL;
        int b,c,i,j,ct;
        char a[20];
        public:

            node *create();
            void insertp();
            void insertm();
            void delm();
            void delp();
            void dels();
            void display();
            void count();
            void reverse();
            void rev(node *p);
            void concat();

} ;
```

```
node *info::create()
{ node *p=new(struct node);
    cout<<"enter name of student ";
    cin>>a;
    strcpy(p->name,a);
     cout<<"\n enter prn no. of student \n";
    cin>>b;
    p->prn=b;
    cout<<"enter student rollno";
    cin>>c;
    p->rollno=c;
    p->next=NULL;
    return  p;
  }
  void info::insertm()
  {
     node *p=create();

    if(head==NULL)
    {    head=p;
    }
   else
   {      temp=head;
          while(temp->next!=NULL)
          {    temp=temp->next;   }
             temp->next=p;
    }

   }
    void info::insertp()
  {
     node *p=create();

    if(head==NULL)
    {    head=p;
    }
   else
   {      temp=head;
           head=p;
             head->next=temp->next;

    }

   }

   void info::display()
   {         if(head==NULL)
              {    cout<<"linklist is empty";
               }
             else
           {
             temp=head;
                     cout<<"     prn     rolln0    NAME   \n";
                     while(temp->next!=NULL)
                     {     cout<<"    \n"<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;
                           temp=temp->next;
                     }
                    cout<<"    "<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;
            }
}
void info::delm()
{  int m,f=0;
   cout<<"\n enter the prn no. of student whose data you want to delete";
    cin>>m;
    temp=head;
    while(temp->next!=NULL)
    {
         if(temp->prn==m)
          {           s->next=temp->next;
                       delete(temp);        f=1;
          }
          s=temp;
          temp=temp->next;
     }      if(f==0)
           {  cout<<"\n sorry memeber not deleted ";  }
 }
 void info::delp()
{     temp=head;
    head=head->next;
       delete(temp);
    }
    void info::dels()
{
    temp=head;
    while(temp->next!=NULL)
    { s=temp;
    temp=temp->next;
    }     s->next=NULL;
       delete(temp);

 }
 void info::count()
 {      temp=head;    ct=0;
        while(temp->next!=NULL)
        {    temp=temp->next; ct++;   }
           ct++;
           cout<<"  Count of members is:"<<ct;

   }
   void info::reverse()
```

```
     {     rev(head);  }
     void info::rev(node *temp)
     {     if(temp==NULL)
           { return; }
           else
           {  rev(temp->next); }
           cout<<"    "<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;
     }

     void info::concat()
    {  int k,j;
       cout<<"enter no. of members in list1";
       cin>>k;
        head=NULL;
       for(i=0;i<k;i++)
       { insertm();
         head1=head;

       } head=NULL;
      cout<<"enter no. of members in list2";
       cin>>j;
       for(i=0;i<j;i++)
       { insertm();
         head2=head;

       } head=NULL;

        temp1=head1;
      while(temp1->next!=NULL)
      {   temp1=temp1->next;   }
            temp1->next=head2;


                            temp2=head1;
                           cout<<"    prn     rolln0    NAME   \n";
                        while(temp2->next!=NULL)
                        {
                           cout<<"\n    "<<temp2->prn<<"    "<<temp2->rollno<<"      "<<temp2-
>name<<"\n";;
                           temp2=temp2->next;
                        }
                           cout<<"\n    "<<temp2->prn<<"    "<<temp2->rollno<<"      "<<temp2-
>name<<"\n";
     }
  int main()
  { info s;
  int i;

          char ch;
       do{
          cout<<"\n choice the options";
          cout<<"\n  1. To insert president    ";
          cout<<"\n 2. To insert member ";
          cout<<"\n 3. To insert secretary ";
          cout<<"\n 4. To delete president ";
          cout<<"\n 5. To delete member ";
          cout<<"\n 6. To delete secretary ";
          cout<<"\n  7. To display data    ";
          cout<<"\n  8. Count of members";
          cout<<"\n 9. To display reverse of string ";
          cout<<"\n 10.To concatenate two strings ";
          cin>>i;
         switch(i)
         {        case 1:   s.insertp();
                              break;
                  case 2:   s.insertm();
                              break;
                  case 3:   s.insertm();
                              break;
                  case 4:   s.delp();
                              break;
                  case 5:   s.delm();
                              break;
                  case 6:  s.dels();
                              break;
                  case 7:  s.display();
                              break;
                  case 8:  s.count();
                              break;
                  case 9:   s.reverse();
                              break;
                  case 10:  s.concat();

                              break;
                  default:  cout<<"\n unknown choice";
          }
            cout<<"\n do you want to continue enter y/Y \n";
            cin>>ch;

       }while(ch=='y'||ch=='Y');

    return 0;
  }
```

OUTPUT :-

```
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary
  4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
  10.To concatenate two strings 1
  enter name of student AJay
  enter prn no. of student
2
enter student rollno20
do you want to continue enter y/Y
Y
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary
  4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
  10.To concatenate two strings 1
  enter name of student Nitin
  enter prn no. of student
3
enter student rollno21
do you want to continue enter y/Y
Y
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary
  4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
  10.To concatenate two strings 3
  enter name of student Sujal
  enter prn no. of student
4
enter student rollno22
do you want to continue enter y/Y
Y
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary
  4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
10.To concatenate two strings 4
do you want to continue enter y/Y
Y
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary 4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
  10.To concatenate two strings 8
  Count of members is:1
 do you want to continue enter y/Y
Y
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary
  4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
  10.To concatenate two strings 7
  prn     rolln0    NAME
    4    22    Sujal
 do you want to continue enter y/Y
Y
choice the options
  1. To insert president
  2. To insert member
  3. To insert secretary
  4. To delete president
  5. To delete member
  6. To delete secretary
  7. To display data
  8. Count of members
  9. To display reverse of string
  10.To concatenate two strings 10
  enter no. of members in list11
  enter name of student Sujal
  enter prn no. of student
```

```
3
enter student rollno22
enter no. of members in list21
enter name of student nitin
enter prn no. of student
2
enter student rollno20
prn     rolln0    NAME

3    22      Sujal

   2    20      nitin

 do you want to continue enter y/Y
y
```

PRACTICAL NO : 09 (C-20)

TITLE : The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 10 rows and 7 seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with. Use array to store pointers (Head pointer) to each row. On demand a) The list of available seats is to be displayed b) The seats are to be booked c) The booking can be cancelled.

Main program :

```
#include<iostream>
 using namespace std;
 class node
 {
  public:
  int seat_no;
  char status;
  node *prev,*next;
 };
 class ticket
 {
  public:
  node *head[10],*last[10];
  ticket()
  {
   for(int i=0;i<10;i++)
   {
    head[i]=NULL;
    last[i]=NULL;
   }
  }
  public:
  void create_bookingsys();
  void display();
  void booking();
  void cancle_booking();
 };
 void ticket::create_bookingsys()
 {
  int i,j;
  for(i=0;i<10;i++)
  {
   for(j=0;j<7;j++)
   {
```

```
   node *ptr;
   ptr=new node();
   ptr->prev=ptr->next=NULL;
   ptr->seat_no=j+1;
   ptr->status='F';
   if(head[i]==NULL)
   {
    head[i]=ptr;
    last[i]=ptr;
   }
   else
   {
    last[i]->next=ptr;
    ptr->prev=last[i];
    ptr->next=head[i];
    last[i]=ptr;
   }
  }
  head[i]->prev=last[i];
 }
}
void ticket::display()
{
 node  *temp;
 for(int i=0;i<10;i++)
 {
  temp=head[i];
  cout<<"ROW NO"<<" "<<i+1<<"->"<<"\t";
  do
  {
   cout<<temp->seat_no<<","<<temp->status<<"\t";
   temp=temp->next;
  }while(temp!=head[i]);
  cout<<"\n";
  }
}
void ticket::booking()
{
 int flag =0;
 int row,seatno,count=0;
 node *temp;
 cout<<"ENTER THE ROW NO";
 cin>>row;
 cout<<"Enter the no of seats";
 cin>>seatno;
 temp=head[row-1];
 do
 {
  if(temp->status=='F')
   {
    count++;
   }
  else if(temp->status=='B')
   {
    count=0;
   }
  if(count==seatno)
    {
     temp=head[row-1];
     if(temp->status!='F')
     {
       while(temp->status!='F')
       {
        temp=temp->next;
       }
       for(int i=0;i<seatno;i++)
       {
        temp->status='B';
        temp=temp->next;
       }
      }
     else
     {
      for(int i=0;i<seatno;i++)
      {
       temp->status='B';
       temp=temp->next;
      }
     }
   }
   temp=temp->next;
  }while(temp!=head[row-1]);
}
void ticket::cancle_booking()
{
 int row,seatno,loc;
 node *temp;
 cout<<"Enter the row from where you want to cancle booking"<<endl;
 cin>>row;
 cout<<"Enter the location for cancling booking"<<endl;
 cin>>loc;
 cout<<"Enter the no of seats are to be cancled"<<endl;
 cin>>seatno;
 temp=head[row-1];
 for(int i=0;i<loc-1;i++)
 {
  temp=temp->next;
 }
 for(int i=0;i<seatno;i++)
 {
  temp->status='F';
  temp=temp->next;
 }
}
```

```
int main()
{
 int ch;
 ticket c;
 do
 {
 cout<<"ENTER YOUR CHOICE:1.createbookingsy\n2.display\n3.booking\n4.canclebooking\n5.exit";
 cin>>ch;

 switch (ch)
 {
  case 1:
        c.create_bookingsys();
        c.display();
        break;
  case 2:
        c.display();
        break;
  case 3:
        c.booking();
        c.display();
        break;
  case 4:
        c.cancle_booking();
        c.display();
        break;
  case 5:

       break;
 }
 }while(ch!=5);

 return 0;
}
```

OUTPUT :-

```
ENTER YOUR CHOICE:1.createbookingsy
2.display
3.booking
4.canclebooking
5.exit1
ROW NO 1->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 2->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 3->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 4->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 5->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 6->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 7->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 8->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 9->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 10->        1,F    2,F    3,F    4,F    5,F    6,F    7,F
ENTER YOUR CHOICE:1.createbookingsy
2.display
3.booking
4.canclebooking
5.exit
3
ENTER THE ROW NO2
Enter the no of seats7
ROW NO 1->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 2->1,B    2,B    3,B    4,B    5,B    6,B    7,B
ROW NO 3->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 4->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 5->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 6->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 7->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 8->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 9->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 10->        1,F    2,F    3,F    4,F    5,F    6,F    7,F
ENTER YOUR CHOICE:1.createbookingsy
2.display
3.booking
4.canclebooking
5.exit4
Enter the row from where you want to cancle booking
2
Enter the location for cancling booking
2
Enter the no of seats are to be cancled
3
ROW NO 1->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 2->1,B    2,F    3,F    4,F    5,B    6,B    7,B
ROW NO 3->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 4->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 5->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 6->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 7->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 8->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 9->1,F    2,F    3,F    4,F    5,F    6,F    7,F
ROW NO 10->        1,F    2,F    3,F    4,F    5,F    6,F    7,F
ENTER YOUR CHOICE:1.createbookingsy
2.display
3.booking
4.canclebooking
5.exit5
```

PRACTICAL NO : 10 (C-21)

TITLE : Write C++ program for storing appointment schedule for day. Appointments are

booked randomly using linked list. Set start and end time and min and max duration for visit slot. Write

functions for- A) Display free slots B) Book appointment C) Sort list based on time D) Cancelappointment (

check validity, time bounds, availability) E) Sort list based on time using pointer manipulation.

Main program :

```
#include<iostream>
#include<string.h>
using namespace std;
int nodes;
struct appoint
{
 int status;
 char start[10];
 char end[10];
 char max[10];
 char min[10];
 struct appoint *Next;
}*head;

void create_app()
{
    int i ;
    appoint *temp, *p;

    head = NULL;

    cout<<"\n\n How many Appointments";
    cin>>nodes;

    for(i=0; i<nodes; i++)
    {
     cout<<" NEW APPOINTMENT ";
      temp = new(struct appoint); //Step 1: Allocate Memory
    temp->status=0;
```

```
       cout<<"\n\t Enter Start Time: "; //Step 2: Store data and address
       cin>>temp->start;
       cout<<"\n\t Enter End Time: "; //Step 2: Store data and address
       cin>>temp->end;
       cout<<"\n\t Enter min Time: "; //Step 2: Store data and address
       cin>>temp->min;
       cout<<"\n\t Enter max Time: "; //Step 2: Store data and address
       cin>>temp->max;

       temp->Next = NULL;

       if(head == NULL)  //Step 3: Attach node in linked List
       {
          head = temp;
          p = head;
       }
       else
       {
          p->Next = temp;
          p = p->Next;
       }

    }

}

void display_SLL()
{
  appoint *p;

    p = head;
    cout<<"Status\tStart Time\tEnd Time\tMin Time\tMax Time\n";
    while(p != NULL)
    {
    if(p->status==0)
    {
    cout<<"Free";
    }
    else
    {
     cout<<"Booked";
 }
       cout<<"\t\t"<<p->start<<"\t\t"<<p->end<<"\t\t"<<p->min<<"\t\t"<<p->max<<"\t\n";
       p = p->Next;
    }
```

```
}
void book_app()
{
char time[10];
struct appoint *p;
cout<<"\n\n\tEnter The Time Slot";
cin>>time;
p=head;
 while(p!=NULL)
 {
  if(strcmp(time,p->start) == 0)
  {
    if(p->status == 0)
    {
     p->status=1;
     cout<<"Your Appointment Is Booked\n\n";
    }
    else
    cout<<"Appointment slot is Busy\n\n";
    break;
  }
  else
   p=p->Next;
 }

 if(p==NULL)
 cout<<"\n\n Appointment slot is Not available\n\n";
 display_SLL();

}

void cancel_app()
{
char time[10];
struct appoint *p;
cout<<"\n\n\tEnter Cancellation Time";
cin>>time;
p=head;
 while(p!=NULL)
 {
  if(strcmp(time,p->start) == 0)
  {
   if(p->status == 1)
    {
     p->status=0;
     cout<<"Your Appointment Is Cancelled\n\n";
    }
    else
    cout<<"Appointment slot is Busy\n\n";
    break;
   }
   else
   p=p->Next;
 }

 if(p==NULL)
 cout<<"\n\n Appointment slot is Not available\n\n";
 display_SLL();


}

void sort_app()
{
char str[10];
struct appoint *p;
int i;
for(i=0;i<nodes-1;i++)
 {
  p = head;
  while(p->Next!=NULL)
  {
   if(strcmp(p->start,p->Next->start)>0)
   {
    int tmp=p->status;
    p->status=p->Next->status;
    p->Next->status=tmp;

    strcpy(str,p->start);
    strcpy(p->start,p->Next->start);
    strcpy(p->Next->start,str);

    strcpy(str,p->end);
    strcpy(p->end,p->Next->end);
    strcpy(p->Next->end,str);

    strcpy(str,p->min);
    strcpy(p->min,p->Next->min);
    strcpy(p->Next->min,str);

    strcpy(str,p->max);
    strcpy(p->max,p->Next->max);
    strcpy(p->Next->max,str);
   }
   p=p->Next;
  }
 }
  cout<<"\n\nSORTED\n";
  display_SLL();
}


int main()
{
create_app();

display_SLL();

book_app();

cancel_app();

sort_app();

return 0;
}
```

OUTPUT :-

```
How many Appointments4

NEW APPOINTMENT

            Enter Start Time: 1

            Enter End Time: 3

            Enter min Time: 2

            Enter max Time: 2

            NEW APPOINTMENT

            Enter Start Time: 3

            Enter End Time: 5

            Enter min Time: 2

            Enter max Time: 2

            NEW APPOINTMENT
```

```
          Enter Start Time: 5

          Enter End Time: 7

          Enter min Time: 2

          Enter max Time: 2

          NEW APPOINTMENT

          Enter Start Time: 7

          Enter End Time: 10

          Enter min Time: 3

          Enter max Time: 3

          Status  Start Time       End Time        Min Time        Max Time

Free              1                3               2               2

Free              3                5               2               2

Free              5                7               2               2

Free              7                10              3               3



          Enter The Time Slot3

          Your Appointment Is Booked


Status  Start Time       End Time        Min Time        Max Time

Free              1                3               2               2

Booked            3                5               2               2

Free              5                7               2               2

Free              7                10              3               3



          Enter Cancellation Time5

          Appointment slot is Busy


Status  Start Time       End Time        Min Time        Max Time

Free              1                3               2               2
```

Booked	3	5	2	2
Free	5	7	2	2
Free7	10	3	3	

SORTED

Status	Start Time	End Time	Min Time	Max Time
Free	1	3	2	2
Booked	3	5	2	2
Free	5	7	2	2
Free	7	10	3	3

PRACTICAL NO : 11 (D-26)

TITLE : - In any language program mostly syntax error occurs due to unbalancing delimiter such as (),{},[]. Write C++ program using stack to check whether given expression is well parenthesized or not.

Main program :

```
#include <iostream>
using namespace std;
#define size 10

class stackexp
{
   int top;
   char stk[size];
public:
   stackexp()
   {
    top=-1;
   }
   void push(char);
   char pop();
   int  isfull();
   int isempty();
};

void stackexp::push(char x)
{
   top=top+1;
   stk[top]=x;
}

char stackexp::pop()
{
   char s;
   s=stk[top];
   top=top-1;
   return s;
```

```
}

int stackexp::isfull()
{
    if(top==size)
        return 1;
    else
        return 0;
}

int stackexp::isempty()
{
    if(top==-1)
        return 1;
    else
        return 0;
}

int main()
{
    stackexp s1;
    char exp[20],ch;
    int i=0;
    cout << "\n\t!! Parenthesis Checker..!!!!" << endl; // prints !!!Hello World!!!
    cout<<"\nEnter the expression to check whether it is in well form or not : ";
    cin>>exp;
    if((exp[0]==')')||(exp[0]==']')||(exp[0]=='}'))
    {
        cout<<"\n Invalid Expression.....\n";
        return 0;
    }
    else
    {
        while(exp[i]!='\0')
        {
            ch=exp[i];
            switch(ch)
            {
            case '(':s1.push(ch);break;
            case '[':s1.push(ch);break;
            case '{':s1.push(ch);break;
```

```
          case ')':s1.pop();break;
          case ']':s1.pop();break;
          case '}':s1.pop();break;
          }
          i=i+1;
       }
   }
   if(s1.isempty())
   {
      cout<<"\nExpression is well parenthesised...\n";
   }
   else
   {
      cout<<"\nSorry !!! Invalid Expression or not in well parenthesized.... \n";
   }
   return 0;
}
```

OUTPUT:-

```
Enter the expression to check whether it is in well form or not : (a+b(b*c-f))
Expression is well parenthesised...

Enter the expression to check whether it is in well form or not : (a+b(b*c-f)
Sorry !!! Invalid Expression or not in well parenthesized....
```

PRACTICAL NO : 12 (D-27)

TITLE : - Implement C++ program for expression conversion as infix to postfix and itsevaluation using stack based on given conditions:

1. Operands and operator, both must be single character.

2. Input Postfix expression must be in a desired format.

3. Only '+', '-', '*' and '/ ' operators are expected.

Main program :

```
#include<iostream>
using namespace std;
class stack
{
 public:
  char stack_array[50];
  int top;
  stack()
  {
    top=-1;
  }
void push(char symbol)
{  if(full())
      cout<<"\nStack overflow:\n";
    else
    { top=top+1;
      stack_array[top]=symbol;
     }
}
char pop()
{  if(empty())
       return('#');        // Return value '#' indicates stack is empty
     else
       return(stack_array[top--]);
}
int empty()
{    if(top==-1)
```

```
      return(1);
    else
    return(0);
}
int full()
{    if(top==49)
    return(1);
    else
    return(0);
}
private:
  char infix[50];
 char postfix[50];
 public:
    void read()
    {
     cout<<"\nEnter an infix expression:";
     cin>>infix;
    }
int white_space(char symbol)
  { if(symbol==' ' || symbol=='\t' || symbol=='\0')
        return 1;
     else
       return 0;
  }
void ConvertToPostfix()
 {   int   prev,p;
     char entry;
     p=0;
     for(int i=0;infix[i]!='\0';i++)
     {
    if(!white_space(infix[i]))
    { switch(infix[i])
     {
       case '(': push(infix[i]);
                 break;
       case ')': while((entry=pop())!='(')
                 postfix[p++]=entry;
                 break;
       case '+':
       case '-':
       case '*':
       case '/':
       if(!empty())
         { prev=prior(infix[i]);
            entry=pop();
        while(prev<=prior(entry))
        { postfix[p++]=entry;
          if(!empty())
             entry=pop();
          else
             break;
        }
       if(prev>prior(entry))
          push(entry);
       }
         push(infix[i]);
         break;
         default:
         postfix[p++]=infix[i];
         break;
       }
     }
    }
     while(!empty())               //while stack is not empty
      postfix[p++]=pop();
    postfix[p]='\0';
    cout<<"\nThe postfix expression is: "<<postfix<<endl;
  }
int prior(char symbol)
{  switch(symbol)
    { case '/': return(4);         // Precedence of / is 4
     case '*': return(3);         // Precedence of * is 3
     case '+': return(2);         // Precedence of + is 2
     case '-': return(1);         // Precedence of - is 1
     case '(': return(0);         // Precedence of ( is 0
     default: return(-1);
    }
  }
};
int main()
{  char choice='y';
```

```
   stack expr;
   while(choice=='y')
  {expr.read();
   expr.ConvertToPostfix();
   cout<<"\n\nDo you want to continue ? (y/n): ";
   cin>>choice;
 }
 return 0;
}
```

OUTPUT:-

```
Enter an infix expression:((a+b)*(c-(d/e)))
The postfix expression is: ab+cde/-*


Do you want to continue ? (y/n): n
```

PRACTICAL NO : 13 (E-29)

TITLE:- Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating jobqueue.Write functions to add job and delete job from queue.

Main program :

```
#include <iostream>
using namespace std;
int queue[100], n = 100, front = - 1, rear = - 1;
void Insert() {
  int val;
  if (rear == n - 1)
  cout<<"Queue Overflow"<<endl;
  else {
    if (front == - 1)
    front = 0;
    cout<<"Insert the element in queue : "<<endl;
    cin>>val;
    rear++;
    queue[rear] = val;
  }
}
void Delete() {
  if (front == - 1 || front > rear)
    {cout<<"Queue Underflow ";
    return ;
  } else {
    cout<<"Element deleted from queue is : "<< queue[front] <<endl;
    front++;;
  }
}
void Display()
  { if (front == -
  1)
  cout<<"Queue is empty"<<endl;
  else {
    cout<<"Queue elements are : ";
    for (int i = front; i <= rear; i++)
    cout<<queue[i]<<" ";
      cout<<endl;
  }
}
int main() {
  int ch;
  cout<<"1) Insert element to queue"<<endl;
  cout<<"2) Delete element from queue"<<endl;
  cout<<"3) Display all the elements of queue"<<endl;
  cout<<"4) Exit"<<endl;
  do {
    cout<<"Enter your choice : "<<endl;
    cin>>ch;
    switch  (ch)
      { case 1:
      Insert();break;
      case 2: Delete();
      break;
      case 3: Display();
      break;
      case 4: cout<<"Exit"<<endl;
      break;
      default: cout<<"Invalid choice"<<endl;
    }
  } while(ch!=4);
  return 0;
}
```

Output:-

```
1) Insert element to queue

2) Delete element from queue

3) Display all the elements of queue

4) Exit

Enter your choice :

1

Insert the element in queue :

5

Enter your choice :

1

Insert the element in queue :

3

Enter your choice :

1

Insert the element in queue :
```

```
7
Enter your choice :
1
Insert the element in queue :
9
Enter your choice :
2
Element deleted from queue is : 5
Enter your choice :
2
Element deleted from queue is : 3
Enter your choice :
3
Queue elements are : 7 9
Enter your choice :4
```

PRACTICAL NO : 14 (E-31)

TITLE:- A double-ended queue(deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate deque with functions to add and delete elements from eitherend of the deque.

Main program :

```
#include<iostream>
using namespace std;
#define SIZE 10
class dequeue {
   int a[20],f,r;
   public:
      dequeue();
      void insert_at_beg(int);
      void insert_at_end(int);
      void delete_fr_front();
      void delete_fr_rear();
      void show();
};
dequeue::dequeue()
   {f=-1;
   r=-1;
}
void dequeue::insert_at_end(int i)
   {if(r>=SIZE-1) {
      cout<<"\n insertion is not possible, overflow!!!!";
   } else {
      if(f==-1)
         {f++;
         r++;
      } else {
         r=r+1;
      }
      a[r]=i;
      cout<<"\nInserted item is"<<a[r];
   }
}
void dequeue::insert_at_beg(int i)
   {if(f==-1) {
      f=0;
      a[++r]=i;
      cout<<"\n inserted element is:"<<i;
   } else if(f!=0) {
      a[--f]=i;
      cout<<"\n inserted element is:"<<i;
   } else {
      cout<<"\n insertion is not possible, overflow!!!";
   }
}
void dequeue::delete_fr_front()
   {if(f==-1) {
      cout<<"deletion is not possible::dequeue is empty";
      return;
   }
   else {
      cout<<"the deleted element is:"<<a[f];
      if(f==r) {
         f=r=-1;
         return;
      } else
         f=f+1;
      }
   }
   void dequeue::delete_fr_rear()
      {if(f==-1) {
         cout<<"deletion is not possible::dequeue is empty";
         return;
      }
      else {
         cout<<"the deleted element is:"<<a[r];
         if(f==r) {
            f=r=-1;
         } else
            r=r-1;
      }
   }
   void dequeue::show()
      {if(f==-1) {
         cout<<"Dequeue is empty";
      } else {
         for(int i=f;i<=r;i++)
            {cout<<a[i]<<" ";
         }
      }
   }
   int main()
      {int c,i;
      dequeue d;
      Do//perform switch opeartion {
```

```
      cout<<"\n 1.insert at beginning";
      cout<<"\n 2.insert at end";
      cout<<"\n 3.show";
      cout<<"\n 4.deletion from front";
      cout<<"\n 5.deletion from rear";
      cout<<"\n 6.exit";
      cout<<"\n enter your choice:";
      cin>>c;
      switch(c)
         {case 1:
            cout<<"enter the element to be inserted";
            cin>>i;
            d.insert_at_beg(i);
         break;
         case 2:
            cout<<"enter the element to be inserted";
            cin>>i;
            d.insert_at_end(i);
         break;
         case 3:
            d.show();
         break;
         case 4:
            d.delete_fr_front();
         break;
         case 5:
            d.delete_fr_rear();
         break;
         case 6:
            exit(1);
         break;
         default:
            cout<<"invalid choice";
         break;
      }
   } while(c!=7);
}
```

Output:-

```
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
```

```
6.exit
enter your choice:4
deletion is not possible::dequeue is empty
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:5
deletion is not possible::dequeue is empty
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:1
enter the element to be inserted7

inserted element is:7
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:1
enter the element to be inserted6

insertion is not possible, overflow!!!
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:1
enter the element to be inserted4

insertion is not possible, overflow!!!
1.insert at beginning
```

```
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:2
enter the element to be inserted6

Inserted item is6
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:2
enter the element to be inserted4

Inserted item is4
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:3
7 6 4
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:4
the deleted element is:7
1.insert at beginning
2.insert at end
3. show
4. deletion from front
5.deletion from rear
6.exit
enter your choice:5
```

```
the deleted element is:4
1.insert at beginning
2.insert at end
3. show
4. deletion from front
5.deletion from rear
6.exit
enter your choice:1
enter the element to be inserted7

inserted element is:7
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:3
7 6
1.insert at beginning
2.insert at end
3.show
4.deletion from front
5.deletion from rear
6.exit
enter your choice:6
```

PRACTICAL NO : 15 (E-32)

TITLE:- Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the systemusing circular queue using array

Main program :

```
#include <iostream>
using namespace std;

class CircularQueue
{private:
    int front, rear, size;
    int* queue;
public:
    CircularQueue(int size)
        {this->size = size;
        queue = new int[size];
        front = rear = -1;
    }

    ~CircularQueue()
        {delete[] queue;
    }

    bool isEmpty()
        { return front == -
        1;
    }

    bool isFull() {
        return (front == 0 && rear == size - 1) || (front == rear + 1);
    }

    void enqueue(int orderNumber)
        {if (isFull()) {
            cout << "Queue is full. Cannot add more orders." << endl;
        } else {
         if (rear == -1)
            { front = rear =
            0;
         } else {
            rear = (rear + 1) % size;
         }
         queue[rear] = orderNumber;
         cout << "Order " << orderNumber << " placed successfully." << endl;
      }
   }

   void dequeue()
      {if (isEmpty())
      {
         cout << "Queue is empty. No orders to serve." << endl;
      } else {
         int servedOrder = queue[front];
         if (front == rear) {
            front = rear = -1;
         } else {
            front = (front + 1) % size;
         }
         cout << "Order " << servedOrder << " served." << endl;
      }
   }
};

int main() {
   int M; // Maximum number of orders
   cout << "Enter the maximum number of orders the pizza parlor can accept: ";
   cin >> M;

   CircularQueue queue(M);

   int choice, orderNumber;
   do {
      cout << "\nPizza Parlor Menu:\n";
      cout << "1. Place an order\n";
      cout << "2. Serve an order\n";
      cout << "3. Exit\n";
      cout << "Enter your choice: ";
      cin >> choice;
```

```
        switch (choice)
            {case 1:
                cout << "Enter order number: ";
                cin >> orderNumber;
                queue.enqueue(orderNumber);
                break;
            case 2:
                queue.dequeue();
                break;
            case 3:
                cout << "Exiting the pizza parlor. Have a nice day!\n";
                break;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != 3);

    return 0;
}
```

Output:-

```
Enter the maximum number of orders the pizza parlor can accept: 6
Pizza Parlor Menu:
1. Place an order
2. Serve an order
3. Exit
Enter your choice: 1
Enter order number: 7
Order 7 placed successfully.

Pizza Parlor Menu:
1. Place an order
2. Serve an order
3. Exit
Enter your choice: 2
Order 7 served.

Pizza Parlor Menu:
1. Place an order
2. Serve an order
```

```
3. Exit
Enter your choice: 3
Exiting the pizza parlor. Have a nice day!
```