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**

CODE :

```
# The average score of class
def average(l):
   sum = 0
   cnt = 0
   for i in range(len(l)):
      if l[i] != -999:
         sum += l[i]
         cnt += 1

   avg = sum / cnt
   print("Total Marks are : ", sum)
   print("Average Marks are : {:.2f}".format(avg))


# Highest score in the class

def Maximum(l):
   Max = l[0]
   for i in range(len(l)):
      if l[i] > Max:
         Max = l[i]
   return (Max)


# Lowest score in the class


def Minimum(l):
   # Assign first element in the array which corresponds to marks of first present student
   # This for loop ensures the above condition

   for i in range(len(l)):
      if l[i] != -999:
         Min = l[i]
         break
```

```
    for j in range(i + 1, len(l)):
        if l[j] != -999 and l[j] < Min:
            Min = l[j]
    return (Min)


# Count of students who were absent for the test

def absentCnt(l):
    cnt = 0
    for i in range(len(l)):
        if l[i] == -999:
            cnt += 1
    return (cnt)


# Display mark with highest frequency
# refeence link : https://www.youtube.com/watch?v=QrIXGqvvpk4&t=422s
def maxFrequency(l):
    i = 0
    Max = 0
    print(" Marks ----> frequency count ")
    for ele in l:
        if l.index(ele) == i:
            print(ele, "---->", l.count(ele))
            if l.count(ele) > Max:
                Max = l.count(ele)
                mark = ele
        i += 1
    return (mark, Max)


# Input the number of students and their corresponding marks in FDS

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

flag = 1
while flag == 1:
    print("/*************MENU**************/")
    print("1. The average score of class ")
    print("2. Highest score and lowest score of class ")
    print("3. Count of students who were absent for the test ")
    print("4. Display mark with highest frequency ")
    print("5. Exit ")
    choice = int(input("Enter your choice : "))

    if choice == 1:
        average(marksInFDS)

    elif choice == 2:
        print("Highest score in the class is : ", Maximum(marksInFDS))
        print("Lowest score in the class is : ", Minimum(marksInFDS))

    elif choice == 3:
        print("Count of students who were absent for the test is : ", absentCnt(marksInFDS))

    elif choice == 4:
        mark, count = maxFrequency(marksInFDS)
        print("Highest frequency of marks {0} is {1} ".format(mark, count))

    else:
        print("Wrong choice")
        flag = 0
```

```
/**********************OUTPUT*********************/
Enter marks of Student 1 : 87
Enter marks of Student 2 : 68
Enter marks of Student 3 : -999
Enter marks of Student 4 : 90
Enter marks of Student 5 : 68
/*************MENU**************/
1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test
4. Display mark with highest frequency
5. Exit
Enter your choice : 1
Total Marks are :  313
Average Marks are : 78.25
/*************MENU**************/
1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test
4. Display mark with highest frequency
5. Exit
Enter your choice : 2
Highest score in the class is :  90
Lowest score in the class is :  68
```

```
/*************MENU**************/
1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test
4. Display mark with highest frequency
5. Exit
Enter your choice : 3
Count of students who were absent for the test is :  1
/*************MENU**************/
1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test
4. Display mark with highest frequency
5. Exit
Enter your choice : 4
 Marks ----> frequency count
87 ----> 1
68 ----> 2
-999 ----> 1
90 ----> 1
Highest frequency of marks 68 is 2
/*************MENU**************/
1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test
4. Display mark with highest frequency
5. Exit
Enter your choice : 5
Wrong choice
```

Write a Python program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following: D 100 W 200 (Withdrawal is not allowed if balance is going negative. Write functions for withdraw and deposit) D means deposit while W means withdrawal.

Suppose the following input is supplied to the program:

D 300, D 300 , W 200, D 100 Then, the output should be: 500

CODE -

```
bal=0
n=int(input("Enter Number of transactions : "))
print("\n")
print("Transaction Format for Widrawal : W 200 and for Deposit : D 200 \n")
for i in range (n):
 x=input("Enter transaction in format : ")
 tran=x.split(" ")
 op=tran[0]
 amt=int(tran[1])
 if op=="D":
  bal+=amt
 elif op=="W":
  if (amt>bal):
   print("Insufficient Amount : ",amt)
  else:
   bal-=amt
 else:
  print("Invalid Choice.")
print("\n")
print("Net A/c. Balance : ",bal)


#Output
#Enter Number of transactions : 4
```

```
#Transaction Format for Widrawal : W 200 and for Deposit : D 200


#Enter transaction in format : D 300

#Enter transaction in format : D 300

#Enter transaction in format : W 200

#Enter transaction in format : D 100



#Net A/c. Balance :  500
```

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

CODE -

```
# Print the matrix

def print_matrix(matrix):
    for i in range(len(matrix)):
        for j in range(len(matrix[0])):
            print("\t", matrix[i][j], end=" ")
        print("\n")

# Initialize the matrix to zero

def init_matrix(matrix, m, n):
    matrix = [[0 for j in range(0, n)] for i in range(0, m)]
    return matrix

# Input the Matrix elements
def read_matrix():
    mat = []
    r = int(input("Enter number of rows in First Matrix: "))
    c = int(input("Enter number of columns in First Matrix : "))
    # in python initilization is needed before indexing.
    mat = init_matrix(mat, r, c)  # matrix 1 initialization with 0s
    for i in range(0, r):
        for j in range(0, c):
            mat[i][j] = int(input("Enter an element : "))
    return mat, r, c

# Addition of two matrices

def mat_add(m1, m2, res, m, n):
    for i in range(0, m):
        for j in range(0, n):
            res[i][j] = m1[i][j] + m2[i][j]
    return res

# Subtraction of two matrices
```

```
def mat_sub(m1, m2, res, m, n):
    for i in range(0, m):
        for j in range(0, n):
            res[i][j] = m1[i][j] - m2[i][j]
    return res

# Multiplication of two matrices

def mat_mul(m1, m2, res, r1, c1, c2):
    # for multiplication
    #  i will run throgh each row of matrix1
    for i in range(0, r1):
        # k will run through each column of matrix 1
        for k in range(0, c2):
            # j will run throguh each column of matrix 2
            for j in range(0, c1):
                res[i][k] += m1[i][j] * m2[j][k]
    return res

# Transpose of matrix1

def trans_mat(m1, res, r1, c1):
    for i in range(0, r1):
        for j in range(0, c1):
            res[j][i] = m1[i][j]
    return res

matrix1 = []
matrix2 = []
res_matrix = []
print(" First Matrix : ")
matrix1, r1, c1 = read_matrix()
print(" Second Matrix : ")
matrix2, r2, c2 = read_matrix()

res_matrix = init_matrix(res_matrix, r1, c2)  # matrix for storing result
# print input matrices
print(" matrix 1")
print_matrix(matrix1)
print(" matrix 2")
print_matrix(matrix2)

flag = 1
while flag:
    print("/*********MENU********/ \n")
    print(" 1. Addition of two matrices ")
    print(" 2. Subtraction of two matrices ")
    print(" 3. Multiplication of two matrices ")
    print(" 4. Transpose of a matrix ")
    choice = int(input("Enter Your Choice : "))

    if choice == 1:
        # printing Addition matrix
        if r1 == r2 and c1 == c2:
            print("resultant matrix after addition")
            res_matrix = mat_add(matrix1, matrix2, res_matrix, r1, c1)
            print_matrix(res_matrix)
        else:
            print("Addition can't be performed ")

    elif choice == 2:
        if r1 == r2 and c1 == c2:
            print("resultant matrix after subtraction")
            res_matrix = mat_sub(matrix1, matrix2, res_matrix, r1, c1)
            print_matrix(res_matrix)
        else:
            print("Subtraction can't be performed ")

    elif choice == 3:
        if c1 == r2:
            print("resultant matrix after Multiplication ")
            res_matrix = mat_mul(matrix1, matrix2, res_matrix, r1, c1, c2)
            print_matrix(res_matrix)
        else:
            print("Multiplication can't be performed ")

    elif choice == 4:
        print("Transpose of First Matrix is ")
        res_matrix1 = []
        res_matrix1 = init_matrix(res_matrix1, c1, r1)
        res_matrix = trans_mat(matrix1, res_matrix1, r1, c1)
        print_matrix(res_matrix)

    else:
        print(" Wrong choice ")
        flag = 0

'''
/**************OUTPUT***************/

 First Matrix :
Enter number of rows in First Matrix: 2
Enter number of columns in First Matrix : 2
Enter an element : 1
Enter an element : 2
Enter an element : 3
Enter an element : 4
 Second Matrix :
Enter number of rows in First Matrix: 2
Enter number of columns in First Matrix : 2
Enter an element : 4
Enter an element : 3
Enter an element : 2
Enter an element : 1
 matrix 1
                1         2

                3         4

 matrix 2
                4         3

                2         1

/*********MENU********/

 1. Addition of two matrices
 2. Subtraction of two matrices
 3. Multiplication of two matrices
 4. Transpose of a matrix
Enter Your Choice : 1
resultant matrix after addition
                5         5

                5         5

/*********MENU********/

 1. Addition of two matrices
 2. Subtraction of two matrices
 3. Multiplication of two matrices
 4. Transpose of a matrix
Enter Your Choice : 2
resultant matrix after subtraction
                -3        -1

                1         3

/*********MENU********/

 1. Addition of two matrices
 2. Subtraction of two matrices
 3. Multiplication of two matrices
 4. Transpose of a matrix
Enter Your Choice : 3
resultant matrix after Multiplication
            5          4

            21         16

/*********MENU********/

 1. Addition of two matrices
 2. Subtraction of two matrices
 3. Multiplication of two matrices
 4. Transpose of a matrix
Enter Your Choice : 4
Transpose of First Matrix is
            1          3

            2          4

/*********MENU********/

 1. Addition of two matrices
 2. Subtraction of two matrices
 3. Multiplication of two matrices
 4. Transpose of a matrix
Enter Your Choice : 5
 Wrong choice
```

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

a) Selection Sort

Bubble sort and display top five scores.

CODE:

```
def selection_sort(arr):
    for i in range(len(arr)):
        for j in range(i + 1, len(arr)):
            if (arr[i] > arr[j]):
                arr[i], arr[j] = arr[j], arr[i]
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
arr = [87.2, 72.3, 91.4, 65.8, 83.7]
print ("Given array is:", arr)
selection_sort(arr)
print ("Sorted array in ascending order using Selection Sort is:", arr)
bubble_sort(arr)
print ("Sorted array in ascending order using Bubble Sort is:", arr)
print("Top five scores are:", arr[-5:])
```

OUTPUT:

```
Given array is: [87.2, 72.3, 91.4, 65.8, 83.7]
Sorted array in ascending order using Selection Sort is: [65.8, 72.3, 83.7, 87.2, 91.4]
Sorted array in ascending order using Bubble Sort is: [65.8, 72.3, 83.7, 87.2, 91.4]
Top five scores are: [65.8, 72.3, 83.7, 87.2, 91.4]
```

Write a Python program to store second year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using

a) Insertion sort

b) Shell Sort and display top five scores

CODE:

```
def insertion_sort(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
def shell_sort(arr):
    interval = len(arr) // 2
    while interval > 0:
        for i in range(interval, len(arr)):
            temp = arr[i]
            j = i
            while j >= interval and temp < arr[j - interval]:
                arr[j] = arr[j - interval]
                j -= interval
            arr[j] = temp
        interval //= 2
arr = [87.2, 72.3, 91.4, 65.8, 83.7]
print ("Given array is:", arr)
insertion_sort(arr)
print ("Sorted array in ascending order using Insertion Sort is:", arr)
shell_sort(arr)
```

```
print ("Sorted array in ascending order using Shell Sort is:", arr)
print("Top five scores are:", arr[-5:])
```

OUTPUT:

```
Given array is: [87.2, 72.3, 91.4, 65.8, 83.7]
Sorted array in ascending order using Insertion Sort is: [65.8, 72.3, 83.7, 87.2, 91.4]
Sorted array in ascending order using Shell Sort is: [65.8, 72.3, 83.7, 87.2, 91.4]
Top five scores are: [65.8, 72.3, 83.7, 87.2, 91.4]
```

Write Python program to store 10th class percentage of students in array. Write function for sorting array of floating point numbers in ascending order using radix sort and display top five scores

CODE:

```
import array as arr


# Accept the % marks of the students

def accept_perc():
    a = arr.array('f', [])
    no_stud = int(input("Enter the number of Students : "))
    for i in range(0, no_stud):
        a.append(float(input("Enter the First Year % of Student[{0}] : ".format(i))))
    return a


# Print the % marks of the Students

def print_perc(a):
    for i in range(0, len(a)):
        print("\t {0:.2f}".format(a[i]), end=" ")
    print()


# Shell Sort

def shell_sort(a):
    # Start with a big gap, then reduce the gap
    n = len(a)
    gap = n // 2

    # Do a gapped insertion sort for this gap size.
    # The first gap elements a[0..gap-1] are already in gapped
    # order keep adding one more element until the entire array
    # is gap sorted
    while gap > 0:

        for i in range(gap, n):

            # add a[i] to the elements that have been gap sorted
            # save a[i] in temp and make a hole at position i
            temp = a[i]

            # shift earlier gap-sorted elements up until the correct
            # location for a[i] is found
            j = i
            while j >= gap and a[j - gap] > temp:
                a[j] = a[j - gap]
                j -= gap

                # put temp (the original a[i]) in its correct location
            a[j] = temp
        gap //= 2
    return a


# Insertion sort

def ins_sort(a):
    # Traverse through 1 to len(a)
    for i in range(1, len(a)):

        key = a[i]

        # Move elements of a[0..i-1], that are
        # greater than key, to one position ahead
        # of their current position
        j = i - 1
        while j >= 0 and key < a[j]:
            a[j + 1] = a[j]
            j -= 1
        a[j + 1] = key
    return a


# Top 5 Score

def top_five(a):
    print("Top five score are : ")
    cnt = len(a)

    if cnt < 5:
        start, stop = cnt - 1, -1  # stop set to -1 as we want to print the 0th element
    else:
        start, stop = cnt - 1, cnt - 6

    for i in range(start, stop, -1):
        print("\t {0:.2f}".format(a[i]), end=" ")


# Driver program
if __name__ == "__main__":
    unsort_A = arr.array('f', [])
    ins_sort_A = arr.array('f', [])
    shell_sort_A = arr.array('f', [])
    flag = 1

    while flag == 1:
        print("\n 1. Accept array elements \n 2. Display the Elements \n 3. Insertion Sort \n 4. Shell Sort
\n 5. exit")
        choice = int(input("Enter your choice : "))

        if choice == 1:
            unsort_A = accept_perc()

        elif choice == 2:
            print_perc(unsort_A)

        elif choice == 3:
            print("Elements after sorting using Insertion Sort :")
            ins_sort_A = ins_sort(unsort_A)
            print_perc(ins_sort_A)
            top_five(ins_sort_A)

        elif choice == 4:
            print("Elements after sorting using Shell Sort :")
            shell_sort_A = shell_sort(unsort_A)
            print_perc(shell_sort_A)
            top_five(shell_sort_A)

        else:
            print("Wrong choice")
            flag = 0
```

OUTPUT:

```
 1. Accept array elements
 2. Display the Elements
 3. Insertion Sort
 4. Shell Sort
 5. exit
Enter your choice : 1
Enter the number of Students : 7
Enter the First Year % of Student[0] : 90.4
Enter the First Year % of Student[1] : 67.78
Enter the First Year % of Student[2] : 84.64
Enter the First Year % of Student[3] : 89.56
Enter the First Year % of Student[4] : 45.23
Enter the First Year % of Student[5] : 58.88
Enter the First Year % of Student[6] : 47.75

 1. Accept array elements
 2. Display the Elements
 3. Insertion Sort
 4. Shell Sort
 5. exit
Enter your choice : 2
          90.40    67.78    84.64    89.56    45.23    58.88    47.75

 1. Accept array elements
 2. Display the Elements
 3. Insertion Sort
 4. Shell Sort
 5. exit
Enter your choice : 3
Elements after sorting using Insertion Sort :
          45.23    47.75    58.88    67.78    84.64    89.56    90.40
Top five score are :
          90.40    89.56    84.64    67.78    58.88
 1. Accept array elements
 2. Display the Elements
 3. Insertion Sort
 4. Shell Sort
 5. exit
Enter your choice : 4
Elements after sorting using Shell Sort :
          45.23    47.75    58.88    67.78    84.64    89.56    90.40
Top five score are :
          90.40    89.56    84.64    67.78    58.88
 1. Accept array elements
 2. Display the Elements
 3. Insertion Sort
 4. Shell Sort
 5. exit
Enter your choice : 5
Wrong choice

****************OUTPUT******************/
```

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

CODE:

```
#include <iostream>
   #include<stdlib.h>
   using namespace std;
   class node
   { public:
   node* next;
   node* prev;
   int seat;
   string id;
   int status;
   };
   class cinemax
   {
   public:
   node* head,* tail ,* temp;
   cinemax()
   {
   head=NULL;
   }
   void create_list();
   void display();
   void book();
   void cancel();

   void avail();


   };

   void cinemax::create_list()

   {

   int i=1;

   temp=new node;

   temp->seat=1;

   temp->status=0;

   temp->id="null";

   tail=head=temp;

   for(int i=2;i<=70;i++)

   {

   node *p;

   p= new node;

   p->seat=i;

   p->status=0;

   p->id="null";

   tail->next=p;

   p->prev=tail;

   tail=p;

   tail->next=head;

   head->prev=tail;


   }

   }

   void cinemax::display()

   {

   { int r=1;

   node* temp;
   temp=head;
   int count=0;
   cout<<"\n---------------------------------------------------------------------------\n";
   cout<<" Screen this way \n";
   cout<<"---------------------------------------------------------------------------\n";
   while(temp->next!=head)
   {
   if(temp->seat/10==0)
   cout<<"S0"<<temp->seat<<" :";
   else
   cout<<"S"<<temp->seat<<" :";

   if(temp->status==0)
   cout<<"|___| ";
   else
   cout<<"|_B_| ";
   count++;
   if(count%7==0)
   {
   cout<<endl;
   r++;
   }
   temp=temp->next;
   }
   cout<<"S"<<temp->seat<<" :";
   if(temp->status==0)
   cout<<"|___| ";
   else
   cout<<"|_B_| ";
   }
   }
```

```
void cinemax::book()

{ int x;

string y;

label:

cout<<"\n\n\nEnter seat number to be booked\n";

cin>>x;

cout<<"Enter your ID number\n";

cin>>y;

if(x<1||x>70)

{

cout<<"Enter correct seat number to book (1-70)\n";

goto label;

}

node *temp;

temp=new node;

temp=head;

while(temp->seat!=x)

{

temp=temp->next;

}


if(temp->status==1)

cout<<"Seat already booked!\n";

else{

temp->status=1;

temp->id=y;

cout<<"Seat "<<x<<" booked!\n";

}


}

void cinemax::cancel()
```

```
{
int x;
string y;
label1:
cout<<"Enter seat number to cancel booking\n";
cin>>x;
cout<<"Enter you ID\n";
cin>>y;
if(x<1||x>70)
{
cout<<"Enter correct seat number to cancel (1-70)\n";
goto label1;
}
node *temp;
temp=new node;
temp=head;
while(temp->seat!=x)
{
temp=temp->next;
}
if(temp->status==0)
{
cout<<"Seat not booked yet!!\n";
}
else
{
if(temp->id==y)
{
temp->status=0;
cout<<"Seat Cancelled!\n";
}
```

```
else
cout<<"Wrong User ID !!! Seat cannot be cancelled!!!\n";
}
}
void cinemax::avail()
{
int r=1;
node* temp;
temp=head;
int count=0;
cout<<"\n\n\n\n";
cout<<"\n------------------------------------------------------------------------------------------\n";
cout<<" Screen this way \n";
cout<<"------------------------------------------------------------------------------------------\n";
while(temp->next!=head)
{
{
if(temp->seat/10==0)
cout<<"S0"<<temp->seat<<" :";
else
cout<<"S"<<temp->seat<<" :";
if(temp->status==0)
cout<<"|___| ";
else if(temp->status==1)
cout<<" ";

count++;
if(count%7==0)

{
```

```
cout<<endl;

}

}
temp=temp->next;
}
if(temp->status==0)
{
cout<<"S"<<temp->seat<<" :";

if(temp->status==0)
cout<<"|___| ";
}

}
int main()
{ cinemax obj;
obj.create_list();
int ch;
char c='y';
while(c=='y')
{ obj.display();
cout<<"\n*************************************************\n";
cout<<"          CINEMAX MOVIE THEATRE       \n";
cout<<"*************************************************\n";
cout<<"\nEnter Choice\n1.Current Seat Status\n2.Book Seat \n3.Available Seat\n4.CancelSeat\n";
cin>>ch;
switch(ch)
{
```

```
case 1:obj.display();
break;
case 2: obj.book();
break;
case 3:obj.avail();
break;
case 4: obj.cancel();
break;
default: cout<<"Wrong choice input\n";
}
cout<<"\nDo you want to perform any other operation : (y/n)\n";
cin>>c;
}
return 0;
}
```

Output :

```
pllab0112@pllab0112-ThinkCentre-M70s:~$ g++ cinema.cpp
pllab0112@pllab0112-ThinkCentre-M70s:~$ ./a.out

--------------------------------------------------------------------------------
 Screen this way
--------------------------------------------------------------------------------
S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|
S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|
S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|
S22 :|___| S23 :|___| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|
S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|
S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|

S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|

S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|

S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|

S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
*********************************************

      CINEMAX MOVIE THEATRE

*********************************************


Enter Choice

1.Current Seat Status

2.Book Seat

3.Available Seat

4.CancelSeat

1


-------------------------------------------------------------------------------------

 Screen this way

-------------------------------------------------------------------------------------

S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|

S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|

S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|

S22 :|___| S23 :|___| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|

S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|

S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|

S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|

S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|

S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|

S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|

Do you want to perform any other operation : (y/n)

y
```

```
--------------------------------------------------------------------------------
 Screen this way
--------------------------------------------------------------------------------
S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|
S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|
S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|
S22 :|___| S23 :|___| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|
S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|
S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|
S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|
S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|
S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|
S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
*********************************************
      CINEMAX MOVIE THEATRE
*********************************************


Enter Choice
1.Current Seat Status
2.Book Seat
3.Available Seat
4.CancelSeat
2



Enter seat number to be booked
23
Enter your ID number
S23
Seat 23 booked!
```

```
Do you want to perform any other operation : (y/n)

y


--------------------------------------------------------------------------------

 Screen this way

--------------------------------------------------------------------------------

S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|

S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|

S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|

S22 :|___| S23 :|_B_| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|

S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|

S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|

S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|

S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|

S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|

S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
*********************************************

     CINEMAX MOVIE THEATRE

*********************************************


Enter Choice

1.Current Seat Status

2.Book Seat

3.Available Seat

4.CancelSeat

3




--------------------------------------------------------------------------------
```

```
 Screen this way
-------------------------------------------------------------------------------------
S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|
S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|
S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|
S22 :|___| S23 : S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|
S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|
S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|
S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|
S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|
S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|
S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
Do you want to perform any other operation : (y/n)
y

-------------------------------------------------------------------------------------
 Screen this way
-------------------------------------------------------------------------------------
S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|
S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|
S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|
S22 :|___| S23 :|_B_| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|
S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|
S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|
S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|
S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|
S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|
S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
*********************************************
      CINEMAX MOVIE THEATRE
*********************************************
```

```
Enter Choice

1.Current Seat Status

2.Book Seat

3.Available Seat

4.CancelSeat

4

Enter seat number to cancel booking

23

Enter you ID

S23

Seat Cancelled!


Do you want to perform any other operation : (y/n)

y


-----------------------------------------------------------------------------------

 Screen this way

-----------------------------------------------------------------------------------

S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|

S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|

S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|

S22 :|___| S23 :|___| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|

S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|

S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|

S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|

S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|

S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|

S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
*********************************************

      CINEMAX MOVIE THEATRE
```

```
*********************************************

Enter Choice
1.Current Seat Status
2.Book Seat
3.Available Seat
4.CancelSeat
1

----------------------------------------------------------------------------------
 Screen this way
----------------------------------------------------------------------------------
S01 :|___| S02 :|___| S03 :|___| S04 :|___| S05 :|___| S06 :|___| S07 :|___|
S08 :|___| S09 :|___| S10 :|___| S11 :|___| S12 :|___| S13 :|___| S14 :|___|
S15 :|___| S16 :|___| S17 :|___| S18 :|___| S19 :|___| S20 :|___| S21 :|___|
S22 :|___| S23 :|___| S24 :|___| S25 :|___| S26 :|___| S27 :|___| S28 :|___|
S29 :|___| S30 :|___| S31 :|___| S32 :|___| S33 :|___| S34 :|___| S35 :|___|
S36 :|___| S37 :|___| S38 :|___| S39 :|___| S40 :|___| S41 :|___| S42 :|___|
S43 :|___| S44 :|___| S45 :|___| S46 :|___| S47 :|___| S48 :|___| S49 :|___|
S50 :|___| S51 :|___| S52 :|___| S53 :|___| S54 :|___| S55 :|___| S56 :|___|
S57 :|___| S58 :|___| S59 :|___| S60 :|___| S61 :|___| S62 :|___| S63 :|___|
S64 :|___| S65 :|___| S66 :|___| S67 :|___| S68 :|___| S69 :|___| S70 :|___|
Do you want to perform any other operation : (y/n)
N
```

Second year Computer Engineering class, set A of students like

Vanilla Ice-cream and set B of students like butterscotch ice-cream. Write C/C++ program to store two sets using linked list. compute and display-

Set of students who like either vanilla or butterscotch or both

ii. Set of students who like both vanilla and butterscotch

iii. Set of students who like only vanilla not butterscotch

iv. Set of students who like only butterscotch not vanilla

v. Number of students who like neither vanilla nor butterscotch

CODE:

```
#include<iostream>
using namespace std;
struct SLLNode* createSLL(int cnt, struct SLLNode *head);
void displaySLL(struct SLLNode *head);
void A_U_B();
void A_int_B();
void A_Min_B();
void B_Min_A();
void U_Min_A_U_B();
struct SLLNode
{
char data;
struct SLLNode *next;
}*headU, *headA, *headB;
int main()
{
int i,no;
cout<<"\n\n\t How many Linked Lists: ";
cin>>no;
headU = headA = headB = NULL;
```

```
for(i=1; i<=no; i++)
{
if(i == 1)
{
cout<<"\n\n\t Enter 10 Students of SE Comp : ";
headU = createSLL(10, headU);
cout<<"\n";
displaySLL(headU);
}

if(i == 2)
{
cout<<"\n\n\t Enter 5 Students who like Vanilla Icecreme: ";
headA = createSLL(5, headA);
cout<<"\n";
displaySLL(headA);
}
if(i == 3)
{
cout<<"\n\n\t Enter 5 Students who like Butterscotch Icecreme: ";
headB = createSLL(5, headB);
cout<<"\n";
displaySLL(headB);
}
}
cout<<"\n\n Input Sets:------------------------";
cout<<"\n\n Set 'U': ";
displaySLL(headU);
cout<<"\n\n Set 'A': ";
displaySLL(headA);
cout<<"\n\n Set 'B': ";
```

```
displaySLL(headB);
cout<<"\n\n Output Sets:------------------------";
A_U_B();
A_int_B();
A_Min_B();
B_Min_A();
U_Min_A_U_B();
cout<<"\n\n";
return 0;
}
//........................................................Function to create
Linked List as Sets.
struct SLLNode* createSLL(int cnt, struct SLLNode *head)
{
int i;
struct SLLNode *p, *newNode;
for(i=0; i<cnt; i++)
{
newNode = new(struct SLLNode); // 1. DMA
cout<<"\n\t Enter Student Initial: "; // 2. Data & Address Assignment
cin>>newNode->data;
newNode->next = NULL;
if(head == NULL) // 3. Add node in the list

{
head = newNode;
p = head;
}
else
{
p->next = newNode;
p = p->next;
}
}
return head;
}

//............................................Function to display Linked Lists
as Sets.
void displaySLL(struct SLLNode *head)
{
struct SLLNode *p;
p = head;
while(p != NULL)
{
cout<<" "<<p->data;
p = p->next;
}
}

//.............................................Function for Set A U B .
void A_U_B()
{
int i,j;
char a[10];
struct SLLNode *p, *q;
i = 0; //Index of Resultant Array
p = headA; //pointer to Set 'A'
q = headB; //pointer to Set 'B'
while(p != NULL && q != NULL)
{
if(p->data == q->data)
```

```
{
a[i] = p->data;
i++;
p = p->next;
q = q->next;
}
else
{
a[i] = p->data;
i++;

p = p->next;
}
}
if(p == NULL) //Set 'A' copied completely
{
while(q != NULL) //Copy remaining elements of Set 'B'
{
a[i] = q->data;
i++;
q = q->next;
}
}
if(q == NULL) //Set 'B' copied completely
{
while(p != NULL) //Copy remaining elements of Set 'A'
{
a[i] = p->data;
i++;
p = p->next;
}
```

```
}
cout<<"\n\n\t Set A U B: ";
for(j=0; j < i; j++)
cout<<" "<<a[j];
}
//................................................Function for Set A ^ B .
void A_int_B()
{
int i,j;
char a[10];
struct SLLNode *p, *q;
i = 0; //Index of Resultant Array
p = headA; //pointer to Set 'A'
while(p != NULL)
{
q = headB; //pointer to Set 'B'
while(q != NULL)
{
if(p->data == q->data)
{
a[i] = p->data;
i++;
}
q = q->next;
}
p = p->next;
}
cout<<"\n\n\t Set A ^ B: ";
for(j=0; j < i; j++)

cout<<" "<<a[j];
```

```
}
//............................................Function for Set A - B .
void A_Min_B()
{
int i,j,flag;
char a[10];
struct SLLNode *p, *q;
i = 0; //Index of Resultant Array
p = headA; //pointer to Set 'A'
while(p != NULL)
{
flag = 0;
q = headB; //pointer to Set 'B'
while(q != NULL)
{
if(p->data == q->data)
{
flag = 1;
}
q = q->next;
}
if(flag == 0)
{
a[i] = p->data;
i++;
}
p = p->next;
}
cout<<"\n\n\t Set A - B: ";
for(j=0; j < i; j++)
cout<<" "<<a[j];
```

```
}

//..............................................Function for Set B - A.
void B_Min_A()
{
int i,j,flag;
char a[10];
struct SLLNode *p, *q;
i = 0; //Index of Resultant Array
q = headB; //pointer to Set 'B'
while(q != NULL)
{
flag = 0;
p = headA; //pointer to Set 'A'
while(p != NULL)

{
if(q->data == p->data)
{
flag = 1;
}
p = p->next;
}
if(flag == 0)
{
a[i] = q->data;
i++;
}
q = q->next;
}
cout<<"\n\n\t Set B - A: ";
```

```
for(j=0; j < i; j++)
cout<<" "<<a[j];
}
//...............................................Function for Set U - (A U B).
void U_Min_A_U_B()
{
int i,j,flag;
char a[10];
struct SLLNode *p, *q, *r;
i = 0; //Index of Resultant Array
p = headU; //pointer to Set 'U'
while(p != NULL)
{
flag = 0;
q = headA; //pointer to Set 'A'
r = headB; //pointer to Set 'B'
while(q != NULL)
{
if(p->data == q->data)
{
flag = 1;
}
q = q->next;
}
while(r != NULL)
{
if(p->data == r->data)
{
flag = 1;
}
r = r->next;
}
if(flag == 0)
{
a[i] = p->data;
i++;
}
p = p->next;

}
cout<<"\n\n\t Set U - (A U B): ";
for(j=0; j < i; j++)
cout<<" "<<a[j];
}
/* ----------------OUTPUT-------------------------
student@IOE-L1LAB:~$ g++ Pr_6CD.cpp -o a
student@IOE-L1LAB:~$ ./a
How many Linked Lists: 3
Enter 10 Students of SE Comp :
Enter Student Initial: A
Enter Student Initial: B
Enter Student Initial: C
Enter Student Initial: D
Enter Student Initial: E
Enter Student Initial: F
Enter Student Initial: G
Enter Student Initial: H
Enter Student Initial: I
Enter Student Initial: J
A B C D E F G H I J
Enter 5 Students who like Vanilla Icecreme:
Enter Student Initial: A
Enter Student Initial: B
Enter Student Initial: C
Enter Student Initial: D
Enter Student Initial: E
A B C D E
Enter 5 Students who like Butterscotch Icecreme:
Enter Student Initial: D
Enter Student Initial: E
Enter Student Initial: F

Enter Student Initial: G
Enter Student Initial: H
D E F G H
Input Sets:------------------------
Set 'U': A B C D E F G H I J
Set 'A': A B C D E
Set 'B': D E F G H
Output Sets:------------------------
Set A U B: A B C D E F G H
Set A ^ B: D E
Set A - B: A B C
Set B - A: F G H
Set U - (A U B): I J
```

Implement C++ program for expression conversion as infix to postfix and its evaluation 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.**

CODE

```
#include <iostream>

#include <stack>

using namespace std;


int getPrecedence(char op) {

  switch (op) {

    case '+':

    case '-':

      return 1;

    case '*':

    case '/':

      return 2;

    default:

      return 0;

  }

}

string infixToPostfix(string expression) {

  stack<char> s;

  string postfix;


  for (char ch : expression) {

    if (isalnum(ch)) {

      postfix += ch;

    } else if (ch == '(') {

      s.push(ch);
```

```
        } else if (ch == ')') {
            while (!s.empty() && s.top() != '(') {
                postfix += s.top();
                s.pop();
            }
            if (!s.empty()) {
                s.pop();
            }
        } else {
            while (!s.empty() && getPrecedence(s.top()) >= getPrecedence(ch)) {
                postfix += s.top();
                s.pop();
            }
            s.push(ch);
        }
    }

    while (!s.empty()) {
        postfix += s.top();
        s.pop();
    }

    return postfix;
}

int evaluatePostfix(string expression) {
    stack<int> s;

    for (char ch : expression) {
        if (isdigit(ch)) {
            int operand = ch - '0';
            s.push(operand);
        } else {
            int operand2 = s.top();
            s.pop();
            int operand1 = s.top();
            s.pop();

            switch (ch) {
                case '+':
                    s.push(operand1 + operand2);
                    break;
                case '-':
                    s.push(operand1 - operand2);
                    break;
                case '*':
                    s.push(operand1 * operand2);
                    break;
                case '/':
                    s.push(operand1 / operand2);
                    break;
            }
        }
    }
    return s.top();
}
int main() {
    string expression;
    cout << "Enter the infix expression: ";
    getline(cin, expression);

    string postfix = infixToPostfix(expression);
    cout << "Postfix expression: " << postfix << endl;


    int result = evaluatePostfix(postfix);
    cout << "Evaluation result: " << result << endl;
    return 0;
}
```

OUTPUT :

```
Enter the infix expression: (a+b)*c
Postfix expression: ab+c*
```

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.

CODE :

```
#include <iostream>
#include <stack>
using namespace std;
bool isWellParenthesized(string expression) {
    stack<char> s;
    for (char ch : expression) {
        if (ch == '(') {
            s.push(ch);
        } else if (ch == ')') {
            if (s.empty()) {
                return false;
            }
            s.pop();
        }
    }
    return s.empty();
}
int main() {
    string expression;
    cout << "Enter the expression: ";
    getline(cin, expression);
    if (isWellParenthesized(expression)) {
        cout << "The expression is well parenthesized" << endl;
    } else {
        cout << "The expression is not well parenthesized" << endl;
    }
    return 0;
}
```

OUTPUT:

```
Enter the expression: (a+b)*c

The expression is well parenthesized
```

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 job queue. Write functions to add job and delete job from queue.

CODE:

```
#include <iostream>
using namespace std;
#define MAX 50
class Queue
{
        public:
                int que[MAX], first = -1, last= -1;

        void InsertQueue()
        {
                int val;
                if(last == MAX-1)
                {
                        cout<<"Queue Overflow!!!"<<endl;
                }
                else
                {
                        if(first == -1)
                                first = 0;
                                cout<<"Insert element in Queue : ";
                                cin>>val;
                                last++;
                                que[last] = val;
                }
```

```
	}

	void DeleteQueue()
	{
		if(first == -1 || first>last)
		{
			cout<<"Queue Underflow!!!"<<endl;
			return;
		}
		else
		{
			cout<<"Element Deleted from Queue : "<<que[first]<<endl;
			first++;
		}
	}

	void DisplayQueue()
	{
		if(first == -1)
		{
			cout<<"Queue is Empty!!!"<<endl;
		}
		else
		{
			cout<<"Queue Elements :\n";
			for(int i=first; i<=last; i++)
				cout<<que[i]<<endl;
			cout<<endl;
		}
	}
};
```

```
int main()
{
        Queue Q;
        cout<<"***   Queue   ***\n"<<endl;
        cout<<"1. Insert element in Queue."<<endl;
  cout<<"2. Delete element from Queue."<<endl;
  cout<<"3. Display elements of Queue."<<endl;
  cout<<"4. Exit."<<endl;
  int ch;
  do{
        cout<<"Enter your Choice : ";
        cin>>ch;
        switch(ch)
        {
                case 1:
                                Q.InsertQueue();
                                break;
                        case 2:
                                Q.DeleteQueue();
                                break;
                        case 3:
                                Q.DisplayQueue();
                                break;
                        case 4:
                                cout<<"Exit.";
                                break;
                        default:
                                cout<<"Invalid Choice."<<endl;
                }
        }while(ch != 4);
        return 0;
}



//Output:

/*      ***   Queue   ***
1. Insert element in Queue.
2. Delete element from Queue.
3. Display elements of Queue.
4. Exit.
Enter your Choice : 3
Queue is Empty!!!
Enter your Choice : 1
Insert element in Queue : 99
Enter your Choice : 1
Insert element in Queue : 100
Enter your Choice : 1
Insert element in Queue : 101
Enter your Choice : 1
Insert element in Queue : 102
Enter your Choice : 1
Insert element in Queue : 103
Enter your Choice : 3
Queue Elements :
99
100
101
102
103
```

```
Enter your Choice : 2
Element Deleted from Queue : 99
Enter your Choice : 3
Queue Elements :
100
101
102
103
Enter your Choice : 4
Exit.*
```

Write program to implement a priority queue in C++ using an inorder list to store the items in the queue. Create a class that includes the data items (which should be template) and the priority (which should be int). The inorder list should contain these objects, with operator <= overloaded so that the items with highest priority appear at the start of the list (which will make it relatively easy to retrieve the highest item.)

CODE :

```
#include <iostream>

#include <string>

#include <queue>


using namespace std;


// Create a Comparison Class for our

// integer priority queue

class CompareClass {

  public:

    bool operator () (int a, int b) {

      if (a <= b)

        return true;

      return false;

    }

};


void print_pqueue (priority_queue<int, vector<int>, CompareClass> pq) {

  // Prints the Priority Queue

  priority_queue<int, vector<int>, CompareClass> copy_q = pq;

  cout << "Priority Queue : ";

  while (!copy_q.empty()) {

    cout << copy_q.top() << " ";
```

```
        copy_q.pop();
    }
    cout << "\n";
}

int main() {
    // Program demonstrating use of Priority Queue
    // methods

    // Create an empty priority queue of integers
    priority_queue<int, vector<int>, CompareClass> queue_int;

    // Is the Queue empty now? Yes!
    cout << "Is the Queue empty now? : " << (queue_int.empty() ? "Yes" : "No") << endl;

    // Let's add some elements!
    cout << "Enter number of elements to add...\n";
    int number,ele;
    cin >> number;
    for(int i = 0; i < number; i++)
    {
        cout <<"\n Enter Element : ";
        cin >> ele;
        queue_int.push(ele);
    }



    cout << "Number of elements : " << queue_int.size() << endl;
    cout << "Top element : " << queue_int.top() << endl << endl;

    print_pqueue(queue_int);
    cout << "Popping element from the top...\n\n";
    queue_int.pop();
    print_pqueue(queue_int);


    return 0;
}
```

```
/***********************************OUTPUT***************************
 Is the Queue empty now? : Yes
Enter number of elements to add...
3



 Enter Element : 10



 Enter Element : 20


 Enter Element : 30
Number of elements : 3
Top element : 30


Priority Queue : 30 20 10
Popping element from the top...



Priority Queue : 20 10
```

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 system using circular queue using array.

CODE :

```
#include<iostream>

#include<cstdlib>

using namespace std;


class pizza

{

    int front,rear,q[5];

    public:

        pizza()

        {

           front=-1;

           rear=-1;

        }


        int isfull()

        {

            if((front==0&&rear==4)||front==rear+1)

            {

          return 1;

            }

            else

            {

          return 0;

            }

        }
```

```
int isempty()

{

        if(front==-1&&rear==-1)

        {

            return 1;

        }

        else

        {

            return 0;

        }

}


void add()

{

    if(isfull()==0)

    {

      cout<<"\n Enter the Pizza ID: ";

      if(front==-1&&rear==-1)

      {

          front=0;

          rear=0;

          cin>>q[rear];

      }

      else

      {

          rear=(rear+1)%5;

          cin>>q[rear];

      }

      char c;

    cout<<" Do you want to add another order ? ";
```

```
      cin>>c;
      if(c=='y'||c=='Y')
      add();
    }
    else
    {
      cout<<"\n Orders are full ";
    }

}

void serve()
{
    if(isempty()==0)
    {
      if(front==rear)
      {
        cout<<"\n Order served is : "<<q[front];
        front=-1;
        rear=-1;
      }
      else
      {
        cout<<"\n Order served is : "<<q[front];
        front=(front+1)%5;
      }
    }
    else
    {
      cout<<"\n Orders are empty ";
    }
```

```
}


void display()
{
   if(isempty()==0)
   {
         for(int i=front;i!=rear;i=(i+1)%5)
         {
            cout<<q[i]<<" <- ";
         }
     cout<<q[rear];
   }
   else
   {
   cout<<"\n Orders are empty";
  }
}


void check()
{
      int ch;
      cout<<"\n\nPIZZA PARLOUR !\n\n";
      cout<<"\n 1. Add a Pizza in Queue \n 2. Display the Orders \n 3. Serve a pizza \n 4. Exit
\n Enter your choice : ";
      cin>>ch;
      switch(ch)
      {
            case 1:
                  add();
                  break;
```

```
                case 2:
                    display();
                    break;

                case 3:
                    serve();
                    break;

                case 4:
                    exit(0);

                default:
                    cout<<" Invalid choice ";
                    check();
            }
            char ch1;
            cout<<"\n Do you want to continue? ";
            cin>>ch1;
            if(ch1=='y'||ch1=='Y')
            check();
        }
};

 int main()
 {
     pizza p1;
     p1.check();
     return 0;
 }
```

OUTPUT :

```
PIZZA PARLOUR !

 1. Add a Pizza in Queue
 2. Display the Orders
 3. Serve a pizza
 4. Exit
 Enter your choice : 1

 Enter the Pizza ID: 2
 Do you want to add another order ? n

 Do you want to continue? y


PIZZA PARLOUR !


 1. Add a Pizza in Queue
 2. Display the Orders
 3. Serve a pizza
 4. Exit
 Enter your choice : 2
2
 Do you want to continue? y


PIZZA PARLOUR !
```

```
 1. Add a Pizza in Queue

 2. Display the Orders

 3. Serve a pizza

 4. Exit

 Enter your choice : 3


 Order served is : 2

 Do you want to continue? y



PIZZA PARLOUR !



 1. Add a Pizza in Queue

 2. Display the Orders

 3. Serve a pizza

 4. Exit

 Enter your choice : 4
```