DY Patil Institute of Engineering Management and Research, Pune

Data Structure and Algorithm Lab

Index

Sr. No.	Experiment Number	Date of Performance
1.	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	5/12/2021
2.	Write a python Program for magic square. A magic square is an n * n matrix of the integers 1 to n2 such that the sum of each row, column, and diagonal is the same. The figure given below is an example of magic square for case n=5. In this example, the common sum is 65.	6/12/2021
3.	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.	6/12/2021
4.	a) Write a python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search. A1 A2 b) Write a python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search A3 A4	11/12/2021 14/12/2021
5.	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.	14/12/2021
6.	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 score	14/12/2021

7.	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 one may 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	14/12/2021
	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.	14/12/2021
8.	Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set B of students like butterscotch ice-cream. Write C++ program to store two sets using linked list. compute and displaya) Set of students who like both vanilla and butterscotch b) Set of students who like either vanilla or butterscotch or not both c) Number of students who like neither vanilla nor butterscotch	14/12/2021
9.	A palindrome is a string of character that‘s the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, “Poor Dan is in a droop” is a palindrome, as can be seen by examining the characters “poor danisina droop” and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functionsa) To print original string followed by reversed string using stack b) To check whether given string is palindrome or no	14/12/2021
10.	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. 3. Only '+', '-', '*' and '/ ' operators are expected	14/12/2021
11.	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 beginning of the list (which will make it relatively easy to retrieve the highest item.)	14/12/2021
12.	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 either end of the deque.	14/12/2021

13.	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.	14/12/2021

Program 1: 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

```
import string
def read(n):
    i = 0
    while(i<n):
        num = input()
        if(num.isdigit() or num.upper() == "NA"):
            marks.append(num)
            i = i + 1
        else:
            print("Invalid value. Please Re-enter the same students score. ")
def average():
    s = 0.0
    for i in marks:
        if(i.isdigit()):
            s = s + int(i)
    avg = s/len(marks)
    print("The average score of the class is : ",avg)


def Max_and_Min():
    Max = 0
    Min = 100
    for i in marks:
        if(i.isdigit()):
            if(int(i) >= Max):
                Max = int(i)
            if(int(i) <= Min):
                Min = int(i)
```

```
    print("The Highest marks are : ", Max)
    print("The Lowest marks are : ", Min)

def Absent():
    count = 0
    for i in marks:
        if(i.upper() == "NA"):
            count += 1
    print("The number of students who were absent are : ", count)

def Frequency():
    frequency_1 = 0
    frequency_2 = 0                 # will store the highest frequency.
    high_marks = 0                  #will store the marks with highest frequency.
    for i in marks:
        if(i.isdigit()):
            for j in marks:
                if(j.isdigit()):
                    if(int(i) == int(j)):
                        frequency_1 += 1
            if(frequency_1 > frequency_2):
                frequency_2 = frequency_1
                high_marks = int(i)
            frequency_1 = 0
    if(frequency_2 == 1):
        print("All marks are having equal frequency.")
    else:
        print("The marks with highest frequency is : ",high_marks)
        print("It occurred ",frequency_2," times")
```

```
def main():
    global marks
    marks = []
    ch = 0
    n = int(input("Enter the number of students : "))
    print("Enter the mark of each student and press enter (Note: NA for the students who were absent) :
")
    read(n)
    while(ch != 5):
        print("Enter your choice : ")
        print("1. To calculate the average of the class : ")
        print("2. To display the minimum and maximum marks : ")
        print("3. To calculate the number of students absent for the exam : ")
        print("4. To calculate the marks with highest frequency : ")
        print("5. Exit ")
        ch = int(input())
        if(ch == 1):
            average()
        elif ch == 2:
            Max_and_Min()
        elif ch == 3:
            Absent()
        elif ch == 4:
            Frequency()
        elif ch == 5:
            print("Thank You.")
        else:
            print("Invalid Input. Please Try Again.")
```

```
if __name__ == "__main__":
    main()
```

Output:

```
========== RESTART: F:\PPS Programs\DSL_Program_1_assignment_5_dec.py ==========
Enter the number of students : 4
Enter the mark of each student and press enter (Note: NA for the students who were absent) :
87
NA
37
57
Enter your choice :
1. To calculate the average of the class :
2. To display the minimum and maximum marks :
3. To calculate the number of students absent for the exam :
4. To calculate the marks with highest frequency :
5. Exit
1
The average score of the class is :  45.25
Enter your choice :
1. To calculate the average of the class :
2. To display the minimum and maximum marks :
3. To calculate the number of students absent for the exam :
4. To calculate the marks with highest frequency :
5. Exit
2
The Highest marks are :  87
The Lowest marks are :  37
Enter your choice :
1. To calculate the average of the class :
2. To display the minimum and maximum marks :
3. To calculate the number of students absent for the exam :
4. To calculate the marks with highest frequency :
5. Exit
3
The number of students who were absent are :  1
```

```
The number of students who were absent are :  1
Enter your choice :
1. To calculate the average of the class :
2. To display the minimum and maximum marks :
3. To calculate the number of students absent for the exam :
4. To calculate the marks with highest frequency :
5. Exit
4
All marks are having equal frequency.
Enter your choice :
1. To calculate the average of the class :
2. To display the minimum and maximum marks :
3. To calculate the number of students absent for the exam :
4. To calculate the marks with highest frequency :
5. Exit
5
Thank You.
>>>
```

Result : All The Operations executed successfully.

Program 2: Write a python Program for magic square. A magic square is an n * n matrix of the integers 1 to n2 such that the sum of each row, column, and diagonal is the same. The figure given below is an example of magic square for case n=5. In this example, the common sum is 65.

```
def magic_square(n):
    rows,cols=(n,n)
    li = [[0 for i in range(cols)] for j in range(rows)]

    i = int(n/2)  #row number
    j = n-1       #column number
    li[i][j] = 1
    for x in range(1,n**2):
        i = i - 1
        j = j + 1
        if(i == -1 and j == n):
            i = 0
            j = n - 2
            li[i][j] = x+1
        elif(i == -1):
            i = n - 1
            li[i][j] = x+1
        elif(j == n):
            j = 0
            li[i][j] = x + 1
        elif(li[i][j] != 0):
            j = j - 2
            i = i + 1
            li[i][j] = x + 1
```

```
        else:
            li[i][j] = x + 1
    for row in range(n):
        for col in range(n):
            print(li[row][col],end="\t")
        print()


def main():
    name = []
    li = []
    n = int(input("Enter n for the magic square : "))
    magic_square(n)
if __name__ == "__main__":
    main()
```

Output:

```
========== RESTART: F:\PPS Programs\DSL_Program_
Enter n for the magic square : 6
18      11      4       33      26      25
10      3       32      31      24      17
2       0       30      23      16      9
36      29      22      15      8       1
28      21      14      7       6       35
20      19      12      5       34      27
>>>
```

Result : The Matrix executed successfully.

Program 3: 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.

```
def saddle(li,rows,cols):
    point = -2**31
    flag = 0
    count = 0
    for i in range(rows):
        flag = 0
        index_i = i
        mini = min(li[i])
        index_j = li[i].index(mini)
        for j in range(rows):
            if li[index_i][index_j] >= li[j][index_j]:
                point =  li[index_i][index_j]
            else:
                point = -2**31
                flag = 1
                break
        if flag == 0:
            print("The saddle point is : ",point)
            count += 1
    if count == 0:
        print("No Saddle point exists")
```

```
def main():
    rows = int(input("Enter the number of rows of the matrix : "))
    cols = int(input("Enter the number of columns of the matrix : "))
    i,j = (rows,cols)
    li = [[0 for i in range(cols)] for j in range(rows)]
    for i in range(0,rows):
        for j in range(0,cols):
            print("Enter the ",i+1," row and ",j+1," column : ",end=" ")
            n = int(input())
            li[i][j] = n
    saddle(li,rows,cols)
if __name__ == "__main__":
    main()
```

Output:

```
========== RESTART: F:\PPS Programs\DSL_Program_3_ass
Enter the number of rows of the matrix : 3
Enter the number of columns of the matrix : 3
Enter the  1  row and  1  column :  1
Enter the  1  row and  2  column :  2
Enter the  1  row and  3  column :  3
Enter the  2  row and  1  column :  4
Enter the  2  row and  2  column :  5
Enter the  2  row and  3  column :  6
Enter the  3  row and  1  column :  7
Enter the  3  row and  2  column :  8
Enter the  3  row and  3  column :  9
The saddle point is :  7
>>>
```

Result : The Matrix executed successfully.

Program 4a: Write a python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search. A1 A2

```
import string
def searcher(s,li):
    flag = False
    for i in li:
        if i.lower() == s.lower():
            print("Roll number ",s," attended the program.")
            flag = True
    if not(flag):
        print("Roll number ",s," did not attended the program.")



def main():
    roll_no_li = []
    n = int(input("Enter the number of students who attended the program : "))
    i = 1
    while True:
        print("Enter the roll number of the student ",i," : ",end=" ")
        roll_no = input()
        if roll_no in roll_no_li:
            print("Roll number already entered. Please try again.")
            continue
        roll_no_li.append(roll_no)
```

```
        i = i + 1
        if i > n:
            break
    s = input("Enter the roll number to be searched : ")
    searcher(s,roll_no_li)

if __name__ == "__main__":
    main()
```

Output:

```
========= RESTART: F:\PPS Programs\DSL_Program_4_a_assignment
Enter the number of students who attended the program : 5
Enter the roll number of the student  1  :  54
Enter the roll number of the student  2  :  37
Enter the roll number of the student  3  :  35
Enter the roll number of the student  4  :  78
Enter the roll number of the student  5  :  89
Enter the roll number to be searched : 89
Roll number  89  attended the program.
>>>
```

```
========= RESTART: F:\PPS Programs\DSL_Program_4_a_assignmer
Enter the number of students who attended the program : 5
Enter the roll number of the student  1  :  27
Enter the roll number of the student  2  :  65
Enter the roll number of the student  3  :  3
Enter the roll number of the student  4  :  35
Enter the roll number of the student  5  :  54
Enter the roll number to be searched : 10
Roll number  10  did not attended the program.
>>>
```

Result : The Linear Search executed successfully.

Program 4b: Write a python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search A3 A4

```
import string

def binary_search(li,s):
    n = len(li)
    mid = int(n%2)
    low = 0
    high = n-1
    for i in range(n):
        if li[mid] == s:
            print("The roll number ",s," attended the seminar.")
            return
        elif li[mid] > s:
            high = mid - 1
        elif li[mid] < s:
            low = mid + 1
    print("The roll number ",s," did not attend the seminar.")


def fib(n):
    if n == 0:
        return 0
    elif n == 1 or n == 2:
```

```
        return 1
    a = 0
    b = 1
    while True:
        c = a + b
        if c >= n:
            return c
            break
        a = b
        b = c


def find_k(f_k):
    a = 0
    b = 1
    k = 2
    while True:
        c = a + b
        k += 1
        if f_k == c:
            return k
            break
        a = b
        b = c


def fibonacci_search(li,s):
    n = len(li)
    f_k = fib(n)
    k = find_k(f_k)
    k = int(k)
```

```
    offset = -1
    flag = False
    while(k > 0):
        f_k = fib(k-2)
        i = min(offset+f_k,n-1)
        if s == li[i]:
            print("The roll number ",s," attended the seminar.")
            return
        elif s > li[i]:
            k = k - 1
            offset = i
        elif s < li[i]:
            k = k - 2
    print("The roll number ",s," did not attend the seminar.")



def sort(li):
    l = len(li)
    for i in range(0,l):
        for j in range(i+1,l):
            if(li[i] > li[j]):
                li[i],li[j] = li[j],li[i]
    return li


def main():
    roll_no_li = []
    n = int(input("Enter the number of students who attended the program : "))
    i = 1
    while True:
```

```
        print("Enter the roll number of the student ",i," : ",end=" ")
        roll_no = input()
        if roll_no in roll_no_li:
            print("Roll number already entered. Please try again.")
            continue
        roll_no_li.append(roll_no)
        i = i + 1
        if i > n:
            break
    s = input("Enter the roll number to be searched : ")
    roll_no_li = sort(roll_no_li)
    print("-----------Using Binary Search---------------------")
    binary_search(roll_no_li,s)
    print("-----------Using Fibonacci Search---------------------")
    fibonacci_search(roll_no_li,s)

if __name__ == "__main__":
    main()
```

Output:

```
========= RESTART: F:\PPS Programs\DSL_Program_4_b_assignment_14_
Enter the number of students who attended the program : 5
Enter the roll number of the student  1  :  46
Enter the roll number of the student  2  :  65
Enter the roll number of the student  3  :  22
Enter the roll number of the student  4  :  12
Enter the roll number of the student  5  :  25
Enter the roll number to be searched : 22
-----------Using Binary Search--------------------
The roll number  22  attended the seminar.
-----------Using Fibonacci Search--------------------
The roll number  22  attended the seminar.
>>>
```

```
========= RESTART: F:\PPS Programs\DSL_Program_4_b_assignment
Enter the number of students who attended the program : 5
Enter the roll number of the student  1  :  10
Enter the roll number of the student  2  :  99
Enter the roll number of the student  3  :  2
Enter the roll number of the student  4  :  35
Enter the roll number of the student  5  :  64
Enter the roll number to be searched : 11
-----------Using Binary Search--------------------
The roll number  11  did not attend the seminar.
-----------Using Fibonacci Search--------------------
The roll number  11  did not attend the seminar.
>>>
```

Result : The Fibonacci Search and Binary Search algorithm executed successfully.

Program 5: 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.

```
def partition(name,arr,low,high):
    i = low - 1
    pivot = arr[high]
    for j in range(low,high):
        if arr[j] <= pivot:
            i = i + 1
            arr[i],arr[j] = arr[j],arr[i]
            name[i],name[j] = name[j],name[i]
    arr[i+1],arr[high] = arr[high],arr[i+1]
    name[i+1],name[high] = name[high],name[i+1]
    return (i+1)


def quick_sort(name,li,low,high):
    if low < high:
        pi = partition(name,li,low,high)
        quick_sort(name,li,low,pi-1)
        quick_sort(name,li,pi+1,high)


def main():
    name = []
    li = []
    n = int(input("Enter the total number of students : "))
    for i in range(n):
        s = input("Enter the full name of the student : ")
        mark = float(input("Enter the marks scored : "))
        li.append(mark)
```

```
        name.append(s)
    quick_sort(name,li,0,n-1)
    print("The top five scorers are : ")
    print("{0:^10}||{1:^50}||{2:^50}".format("Rank","Name of the Student","Marks Obtained"))
    if n <= 5:
        for i in range(-1,-n-1,-1):
            print("{0:^10}||{1:^50}||{2:^50}".format(-i,name[i],li[i]))
    else:
        for i in range(-1,-6,-1):
            print("{0:^10}||{1:^50}||{2:^50}".format(-i,name[i],li[i]))


if __name__ == "__main__":
    main()
```

Output:

```
========== RESTART: F:\PPS Programs\DSL_Program_5_assignment_14_dec.py =========
Enter the total number of students : 6
Enter the full name of the student : Pooja
Enter the marks scored : 87.25
Enter the full name of the student : Sunil
Enter the marks scored : 70.24
Enter the full name of the student : Manav
Enter the marks scored : 47.24
Enter the full name of the student : Neel
Enter the marks scored : 88.22
Enter the full name of the student : Vinod
Enter the marks scored : 55.36
Enter the full name of the student : Garima
Enter the marks scored : 98.55
The top five scorers are :
   Rank    ||               Name of the Student               ||                 Marks Obtained
    1      ||                      Garima                     ||                     98.55
    2      ||                       Neel                      ||                     88.22
    3      ||                      Pooja                      ||                     87.25
    4      ||                      Sunil                      ||                     70.24
    5      ||                      Vinod                      ||                     55.36
>>>
```

Result : The Quick Sort algorithm executed successfully.

Program 6: 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 score

```
def insertion_sort(name,li,n):
    for i in range(1,n):
        j = i - 1
        temp_mark = li[i]
        temp_name = name[i]
        while j >= 0 and li[j] > temp_mark:
            li[j+1] = li[j]
            name[j+1] = name[j]
            j = j - 1
        li[j+1] = temp_mark
        name[j+1] = temp_name
    return li,name
def shell_sort(li,name,n):
    gap = n // 2
    while gap > 0 :
        for i in range(gap , n):
            temp_mark = li[i]
            temp_name = name[i]
            j = i
            while j >= gap and li[j-gap] > temp_mark:
                li[j] = li[j-gap]
                name[j] = name[j-gap]
                j -= gap
            li[j] = temp_mark
            name[j] = temp_name
```

```
        gap //= 2
    return li,name



def main():
    name = []
    li = []
    n = int(input("Enter the total number of students : "))
    for i in range(n):
        s = input("Enter the full name of the student : ")
        mark = float(input("Enter the marks scored : "))
        li.append(mark)
        name.append(s)
    print("----------------Using Insertion Sort----------------------------")
    li,name = insertion_sort(name,li,n)
    print("The top five scorers are : ")
    print("{0:^10}||{1:^50}||{2:^50}".format("Rank","Name of the Student","Marks Obtained"))
    if n <= 5:
        for i in range(-1,-n-1,-1):
            print("{0:^10}||{1:^50}||{2:^50}".format(-i,name[i],li[i]))
    else:
        for i in range(-1,-6,-1):
            print("{0:^10}||{1:^50}||{2:^50}".format(-i,name[i],li[i]))
    print("----------------Using Shell Sort----------------------------")
    li,name = shell_sort(li,name,n)
    print("The top five scorers are : ")
    print("{0:^10}||{1:^50}||{2:^50}".format("Rank","Name of the Student","Marks Obtained"))
    if n <= 5:
        for i in range(-1,-n-1,-1):
```

```
        print("{0:^10}||{1:^50}||{2:^50}".format(-i,name[i],li[i]))
    else:
        for i in range(-1,-6,-1):
            print("{0:^10}||{1:^50}||{2:^50}".format(-i,name[i],li[i]))


if __name__ == "__main__":
    main()
```

Output:

```
========== RESTART: F:\PPS Programs\DSL_Program_6_assignment_14_dec.py =========
Enter the total number of students : 5
Enter the full name of the student : Pooja
Enter the marks scored : 88.68
Enter the full name of the student : Sunil
Enter the marks scored : 57.69
Enter the full name of the student : Manav
Enter the marks scored : 98.57
Enter the full name of the student : Garima
Enter the marks scored : 44.65
Enter the full name of the student : Vinod
Enter the marks scored : 78.25
----------------Using Insertion Sort----------------------------
The top five scorers are :
   Rank    ||               Name of the Student                ||                  Marks Obtained
    1      ||                     Manav                       ||                      98.57
    2      ||                     Pooja                       ||                      88.68
    3      ||                     Vinod                       ||                      78.25
    4      ||                     Sunil                       ||                      57.69
    5      ||                     Garima                      ||                      44.65
----------------Using Shell Sort----------------------------
The top five scorers are :
   Rank    ||               Name of the Student                ||                  Marks Obtained
    1      ||                     Manav                       ||                      98.57
    2      ||                     Pooja                       ||                      88.68
    3      ||                     Vinod                       ||                      78.25
    4      ||                     Sunil                       ||                      57.69
    5      ||                     Garima                      ||                      44.65
>>>
```

Result : The Insertion Sort and Shell Sort algorithm executed successfully.

Program 7: 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 one may 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.

```
#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 concatenate two strings 1
enter name of student Pooja

  enter prn no. of student
100
enter student rollno15

 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 concatenate two strings 2
enter name of student Sunil

  enter prn no. of student
160
enter student rollno25
```

```
 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 concatenate two strings 3
enter name of student Manav

  enter prn no. of student
330
enter student rollno48

 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 concatenate two strings 7
     prn      rolln0     NAME

100     15     Pooja
160     25     Sunil     330     48     Manav
 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 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 concatenate two strings 6

 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 concatenate two strings 5

 enter the prn no. of student whose data you want to delete160

 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 concatenate two strings 8
  Count of members is:3
```

```
 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 concatenate two strings 9
enter no. of members in list12
enter name of student Suraj

  enter prn no. of student
150
enter student rollno36
enter name of student Sunil

  enter prn no. of student
698
enter student rollno35
enter no. of members in list22
enter name of student Manav

  enter prn no. of student
150
enter student rollno38
enter name of student Pooja

  enter prn no. of student
970
enter student rollno26
```

```
enter student rollno26
     prn      rolln0     NAME

    150     36       Suraj

    698     35       Sunil

    150     38       Manav

    970     26       Pooja
```

Result : The Linked List executed successfully.

Program 8: Second year Computer Engineering class, set A of students like Vanilla

Ice-cream and set B of students like butterscotch ice-cream. Write

C++ program to store two sets using linked list. compute and displaya) Set of students who like both vanilla and butterscotch b) Set of students who like either vanilla or butterscotch or not both c) Number of students who like neither vanilla nor butterscotch

```
#include <iostream>

using namespace std;


struct node

{

        char name[100];

        int vanilla;

        int butter_scotch;

        struct node *nxt;

};


void insert(struct node **q,char name[],int v,int b)


{

        struct node *temp;

        temp = *q;

        struct node *newnode;

        newnode = (struct node*)malloc(sizeof(struct node));


        strcpy(newnode->name,name);

        newnode->vanilla = v;

        newnode->butter_scotch = b;

        temp->nxt = newnode;

        temp = newnode;
```

```
        *q = temp;

}

void show(struct node *head)
{
        while(head != NULL)
        {
                cout<<head->name<<endl;
                head = head->nxt;
        }

}


void vanilla_and_butter(struct node *head1,struct node *head2)
{
        cout<<"The following students like both vanilla and butterscotch : "<<endl;
        while(head1 != NULL)
        {
                if(head1->vanilla == 1 and head1->butter_scotch == 1)
                {
                        cout<<head1->name<<"\t"<<"A"<<endl;
                }
                head1 = head1->nxt;
        }
        while(head2 != NULL)
        {
                if(head2->vanilla == 1 and head2->butter_scotch == 1)
```

```
                {
                        cout<<head2->name<<"\t"<<"B"<<endl;
                }
                head2 = head2->nxt;
        }
}

void vanilla_or_butter(struct node *head1,struct node *head2)
{
        cout<<"The following students like either vanilla or butterscotch but not both : "<<endl;
        while(head1 != NULL)
        {
                if((head1->vanilla == 1 and head1->butter_scotch == 2) || (head1->vanilla == 2 and
head1->butter_scotch == 1))
                {
                        cout<<head1->name<<"\t"<<"A"<<endl;
                }
                head1 = head1->nxt;
        }
        while(head2 != NULL)
                {
                        if((head2->vanilla == 1 and head2->butter_scotch == 2) || (head2->vanilla == 2
and head2->butter_scotch == 1))
                        {
                                cout<<head2->name<<"\t"<<"B"<<endl;
                        }
                        head2 = head2->nxt;
                }
}
```

```
void neither(struct node *head1,struct node *head2)
{
	cout<<"The following students like neither vanilla nor butterscotch : "<<endl;
		while(head1 != NULL)
		{
			if(head1->vanilla == 2 and head1->butter_scotch == 2)
			{
				cout<<head1->name<<"\t"<<"A"<<endl;
			}
			head1 = head1->nxt;
		}
		while(head2 != NULL)
		{
			if(head2->vanilla == 2 and head2->butter_scotch == 2)
			{
				cout<<head2->name<<"\t"<<"B"<<endl;
			}
			head2 = head2->nxt;
		}
}




int main()
{
```

```
int c,n,v,b;
struct node* head_A = (struct node*)malloc(sizeof(struct node));
struct node* head_B = (struct node*)malloc(sizeof(struct node));
struct node* tail_A = (struct node*)malloc(sizeof(struct node));
struct node* tail_B = (struct node*)malloc(sizeof(struct node));
char name[100];
cout<<"------------------------------------------------------------------------"<<endl;
cout<<"Enter the number of students of division A : "<<endl;
cin >> n;
if(n == 1)
{
        cout<<"Enter the name 1 : "<<endl;
        cin.ignore();
        cin.getline(name,100);
        strcpy(head_A->name,name);
        cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
        cin >> head_A->vanilla;

        cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for No."<<endl;
        cin >> head_A->butter_scotch;


head_A->nxt = NULL;
}
else
{
        cout<<"Enter the name 1 : "<<endl;
        cin.ignore();
        cin.getline(name,100);
```

```
        strcpy(head_A->name,name);
        cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
        cin >> head_A->vanilla;

        cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for No."<<endl;
        cin >> head_A->butter_scotch;

        cout<<"Enter the name 2 : "<<endl;
        cin.ignore();
        cin.getline(name,100);
        strcpy(tail_A->name,name);
        cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
        cin >> tail_A->vanilla;

        cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for No."<<endl;
        cin >> tail_A->butter_scotch;


tail_A->nxt = NULL;



head_A->nxt = tail_A;
for(int i = 0;i<n-2;i++)
{
        cout<<"Enter the name "<<i+3<<" "<<endl;
        cin.ignore();
                                        cin.getline(name,100);
```

```
                                        cout<<"Do you like vanilla ? Press 1 for Yes 2 for
No."<<endl;
                                        cin >> v;

                                        cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for
No."<<endl;
                                        cin >> b;
                insert(&tail_A,name,v,b);
        }
        }
        cout<<"-----------------------------------------------------------------------"<<endl;
        cout<<"-----------------------------------------------------------------------"<<endl;
        cout<<"Enter the number of students of division B : "<<endl;
                        cin >> n;
                        if(n == 1)
                        {
                                cout<<"Enter the name : "<<endl;
                                cin.ignore();
                                cin.getline(name,100);
                                strcpy(head_B->name,name);
                                cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
                                cin >> head_B->vanilla;

                                cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for
No."<<endl;
                                cin >> head_B->butter_scotch;


                        head_B->nxt = NULL;
                        }
```

```
                else
                {
                        cout<<"Enter the name 1 : "<<endl;
                        cin.ignore();
                        cin.getline(name,100);
                        strcpy(head_B->name,name);
                        cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
                        cin >> head_B->vanilla;

                        cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for
No."<<endl;
                        cin >> head_B->butter_scotch;

                        cout<<"Enter the name 2 : "<<endl;
                        cin.ignore();
                        cin.getline(name,100);
                        strcpy(tail_B->name,name);
                        cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
                        cin >> tail_B->vanilla;

                        cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for
No."<<endl;
                        cin >> tail_B->butter_scotch;


                tail_B->nxt = NULL;
```

```
                head_B->nxt = tail_B;
                for(int i = 0;i<n-2;i++)
                {
                        cout<<"Enter the name "<<i+3<<" "<<endl;
                        cin.ignore();
                                                cin.getline(name,100);

                                                cout<<"Do you like vanilla ? Press 1 for Yes 2 for No."<<endl;
                                                cin >> v;

                                                cout<<"Do you like butter-scotch ? Press 1 for Yes 2 for No."<<endl;
                                                cin >> b;
                        insert(&tail_B,name,v,b);
                }
                }
        cout<<"---------------------------------------------------------------------"<<endl;
        do
        {
                cout<<"---------------------------------------------------------------------"<<endl;
                cout<<"Press 1 to know Set of students who like both vanilla and butter scotch"<<endl;
                cout<<"Press 2 to know Set of students who like either vanilla or butter scotch or not both."<<endl;
                cout<<"Press 3 to know Number of students who like neither vanilla nor butter scotch."<<endl;
                cout<<"Press 0 to exit."<<endl;
                cout<<"---------------------------------------------------------------------"<<endl;
                cin >> c;
```

```
            switch(c)
            {
            case 1:
            {
                  vanilla_and_butter(head_A,head_B);
            }
            break;
            case 2:
            {
                  vanilla_or_butter(head_A,head_B);
            }
            break;
            case 3:
            {
                  neither(head_A,head_B);
            }
            break;
            case 0:
            {
                  cout<<"Exited."<<endl;
            }
            break;
            default:
                  cout<<"Invalid Input."<<endl;
            }
      }while(c != 0);

      return 0;
}
```

Output:

```
------------------------------------------------------------
Enter the number of students of division A :
5
Enter the name 1 :
Vinod
Do you like vanilla ? Press 1 for Yes 2 for No.
1
Do you like butter-scotch ? Press 1 for Yes 2 for No.
1
Enter the name 2 :
Sunil
Do you like vanilla ? Press 1 for Yes 2 for No.
2
Do you like butter-scotch ? Press 1 for Yes 2 for No.
1
Enter the name 3
Pooja
Do you like vanilla ? Press 1 for Yes 2 for No.
2
Do you like butter-scotch ? Press 1 for Yes 2 for No.
2
Enter the name 4
Manav
Do you like vanilla ? Press 1 for Yes 2 for No.
1
Do you like butter-scotch ? Press 1 for Yes 2 for No.
2
Enter the name 5
Shanvi
Do you like vanilla ? Press 1 for Yes 2 for No.
1
Do you like butter-scotch ? Press 1 for Yes 2 for No.
1
```

```
------------------------------------------------------------------------
Enter the number of students of division B :
4
Enter the name 1 :
Jagrut
Do you like vanilla ? Press 1 for Yes 2 for No.
1
Do you like butter-scotch ? Press 1 for Yes 2 for No.
1
Enter the name 2 :
Aarti
Do you like vanilla ? Press 1 for Yes 2 for No.
2
Do you like butter-scotch ? Press 1 for Yes 2 for No.
2
Enter the name 3
Heena
Do you like vanilla ? Press 1 for Yes 2 for No.
1
Do you like butter-scotch ? Press 1 for Yes 2 for No.
2
Enter the name 4
Samiksha
Do you like vanilla ? Press 1 for Yes 2 for No.
2
Do you like butter-scotch ? Press 1 for Yes 2 for No.
1
------------------------------------------------------------------------
```

```
------------------------------------------------------------------------
------------------------------------------------------------------------
Press 1 to know Set of students who like both vanilla and butter scotch
Press 2 to know Set of students who like either vanilla or butter scotch or not both.
Press 3 to know Number of students who like neither vanilla nor butter scotch.
Press 0 to exit.
------------------------------------------------------------------------
1
The following students like both vanilla and butterscotch :
Vinod   A
Shanvi  A
Jagrut  B
------------------------------------------------------------------------
Press 1 to know Set of students who like both vanilla and butter scotch
Press 2 to know Set of students who like either vanilla or butter scotch or not both.
Press 3 to know Number of students who like neither vanilla nor butter scotch.
Press 0 to exit.
------------------------------------------------------------------------
2
The following students like either vanilla or butterscotch but not both :
Sunil   A
Manav   A
Heena   B
Samiksha        B
------------------------------------------------------------------------
```

```
------------------------------------------------------------------------
Press 1 to know Set of students who like both vanilla and butter scotch
Press 2 to know Set of students who like either vanilla or butter scotch or not both.
Press 3 to know Number of students who like neither vanilla nor butter scotch.
Press 0 to exit.
------------------------------------------------------------------------
3
The following students like neither vanilla nor butterscotch :
Pooja   A
Aarti   B
------------------------------------------------------------------------
Press 1 to know Set of students who like both vanilla and butter scotch
Press 2 to know Set of students who like either vanilla or butter scotch or not both.
Press 3 to know Number of students who like neither vanilla nor butter scotch.
Press 0 to exit.
------------------------------------------------------------------------
0
Exited.
```

Result : The Linked List executed successfully.

Program 9: A palindrome is a string of character that's the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functionsa) To print original string followed by reversed string using stack b) To check whether given string is palindrome or no

```
#include<iostream>

#include<string.h>

#define max 50

using namespace std;


class STACK

{

        private:

                char a[max];

                int top;


        public:

                STACK()

                {

                        top=-1;

                }


                void push(char);

                void reverse();

                void convert(char[]);

                void palindrome();

};
```

```
void STACK::push(char c)
{
	top++;
	a[top] = c;
	a[top+1]='\0';


}

void STACK::reverse()
{
	char str[max];

	cout<<"\n\nReverse string is : ";

	for(int i=top,j=0; i>=0; i--,j++)
	{
		cout<<a[i];
		str[j]=a[i];
	}

	cout<<endl;
}


void STACK::convert(char str[])
{
	int j,k,len = strlen(str);
```

```
        for(j=0, k=0; j<len; j++)
        {
                if( ( (int)str[j] >= 97 && (int)str[j] <=122 ) || ( (int)str[j] >= 65 && (int)str[j] <=90 ))
                {
                        if( (int)str[j] <=90 )
                        {
                                str[k] = (char)( (int)str[j] + 32 );
                        }else
                        {
                                str[k] = str[j];
                        }

                        k++;
                }
        }
        str[k]='\0';

        cout<<endl<<"Converted String : "<<str<<"\n";
}




void STACK::palindrome()
{
        char str[max];
        int i,j;

        for(i=top,j=0; i>=0; i--,j++)
```

```
	{
		str[j]=a[i];
	}
	str[j]='\0';



	if(strcmp(str,a) == 0)
		cout<<"\n\nString is palindrome...";
	else
		cout<<"\n\nString is not palindrome...";
}



int main()
{
	STACK stack;

	char str[max];
	int i=0;

	cout<<"\nEnter string to be reversed and check is it palindrome or not : \n\n";

	cin.getline(str , 50);

	stack.convert(str);

	while(str[i] != '\0')
	{
		stack.push(str[i]);
```

```
            i++;
        }

        stack.palindrome();

        stack.reverse();

}
```

Output:

```
Enter string to be reversed and check is it palindrome or not :
Poor dan is in a droop
Converted String : poordanisinadroop

String is palindrome...
Reverse string is : poordanisinadroop
```

Result : The Stack executed successfully.

Program 10: 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. 3. Only '+', '-', '*' and '/ ' operators are expected

```
#include<iostream>
#include<conio.h>
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/-*
```

Program to evaluate postfix form:

```
using namespace std;
#include<iostream>
#include<cono.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#define MAX 50

int stack[MAX];
int top,nn;
char s[20];
void push ( char );
char pop ();
void calculate ();

main( )
{
top = -1 ;
cout<<"\nEnter an expression in postfix form : " ;
get (s) ;
calculate() ;
getch( ) ;
```

```
}
void push(char ch)
{
if(top==MAX-1)
{
cout<<\n\tSTACK is over flow”;
}
else
{
top++;
stack[top] = ch;
}
}
char pop()
{
if(top==-1)
{
cout<<”\n\tSTACK is under flow”;
return(-1);
}
else
{
char ch = stack[top];
top--;
return(ch);
}
}
void calculate
{
```

```
int n1,n2,n3,i ;
i = 0;
while ( s[i] )
{
if ( s[i] == ' ' || s[i] == '\t')
{
i++;
continue ;
}
if ( isdigit ( s[i] ) )
{
nn = s[i] – '0' ;
push ( nn ) ;
}
else
{

n1 = pop () ;
n2 = pop () ;
switch ( s[i] )
{
case '+' :
n3 = n2 + n1 ;
break ;
case '-' :
n3 = n2 - n1 ;
break ;
case '/' :
n3 = n2 / n1 ;
```

```
break ;
case '*' :
n3 = n2 * n1 ;
break ;
default :
printf ( "Unknown operator" ) ;
exit ( 1 ) ;
}
push ( n3 ) ;
}
i++ ;
}
cout<<"\nResult is : "<<n3;
}
```

Output:

```
Enter an expression in postfix form : 53+82-*
Result is : 48
```

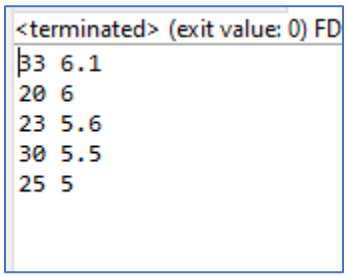
Result : The Stack executed successfully.

Program 11: 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 beginning of the list (which will make it relatively easy to retrieve the highest item.)

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

#define ROW 5
#define COL 2

class Person {
public :
	int age;

	float height;

	Person(int age, float height)
	: age(age), height(height)
	{
	}

};

bool operator<(const Person& p1,const Person& p2)
{

	return p1.height <= p2.height;
}

int main()
{

	priority_queue<Person> Q;

	float arr[ROW] [COL] = { { 30, 5.5 }, { 25, 5 },
				{ 20, 6 }, { 33, 6.1 }, { 23, 5.6 } };

	for (int i = 0; i < ROW; ++i) {

		Q.push(Person(arr[i] [0], arr[i] [1]));
```

```
        }

        while (!Q.empty()) {

                Person p = Q.top();

                Q.pop();

                cout << p.age << " " << p.height << "\n";

        }
        return 0;
}
```

Output:



Result : The Priority Queue executed successfully.

Program 12: 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 either end of the deque.

```
#include<iostream>

//#include

//#include

using namespace std;

#define SIZE 5

// ERROR HANDLINH NOT DOne


//    program is not working correct.

//

class dequeue

{


        int a[10],front,rear,count;


public:

        dequeue();

        void add_at_beg(int);

        void add_at_end(int);

        void delete_fr_front();

        void delete_fr_rear();

        void display();

};
```

```
dequeue::dequeue()
{
        front=-1;
        rear=-1;
        count=0;
}


void dequeue::add_at_beg(int item)
{
        int  i;
        if(front==-1)
        {
                front++;
                rear++;
                a[rear]=item;
                count++;
        }
        else if(rear>=SIZE-1)
        {
                cout<<"\nInsertion is not possible,overflow!!!!";
        }
        else
        {
                for(i=count;i>=0;i--)
```

```
                {
                        a[i]=a[i-1];
                }
                a[i]=item;
                count++;
                rear++;
        }
}



void dequeue::add_at_end(int item)
{

        if(front==-1)
        {
                front++;
                rear++;
                a[rear]=item;
                count++;
        }
        else if(rear>=SIZE-1)
        {
                cout<<"\nInsertion is not possible,overflow!!!";
                return;
        }
        else
```

```
        {
                a[++rear]=item;
        }


}



void dequeue::display()
{

        for(int i=front;i<=rear;i++)
        {
                cout<<a[i]<<" "; }
}


void dequeue::delete_fr_front()
{
        if(front==-1)
        {
                cout<<"Deletion is not possible:: Dequeue is empty";
                return;
        }
        else
        {
```

```
                if(front==rear)
                {
                        front=rear=-1;
                        return;
                }
                cout<<"The deleted element is "<<a[front];
                front=front+1;
        }


}

void dequeue::delete_fr_rear()
{
        if(front==-1)
        {
                cout<<"Deletion is not possible:Dequeue is empty";
                return;
        }
        else
        {
                if(front==rear)
                {
                        front=rear=-1;
                }
                cout<<"The deleted element is "<< a[rear];
                rear=rear-1;
```

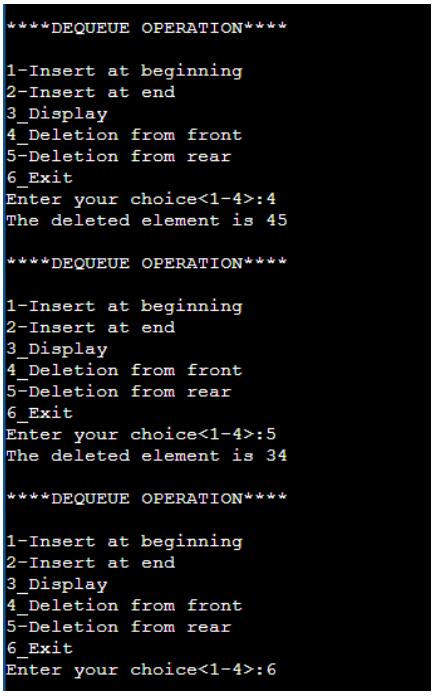
```
        }


}



int main()
{
        int c,item;
        dequeue d1;

        do
        {
                cout<<"\n\n****DEQUEUE OPERATION****\n";
                cout<<"\n1-Insert at beginning";
                cout<<"\n2-Insert at end";
                cout<<"\n3_Display";
                cout<<"\n4_Deletion from front";
                cout<<"\n5-Deletion from rear";
                cout<<"\n6_Exit";
                cout<<"\nEnter your choice<1-4>:";
                cin>>c;

                switch(c)
                {
                case 1:
```

```
            cout<<"Enter the element to be inserted:";
            cin>>item;
            d1.add_at_beg(item);
            break;

case 2:
            cout<<"Enter the element to be inserted:";
            cin>>item;
            d1.add_at_end(item);
            break;

case 3:
            d1.display();
            break;

case 4:
            d1.delete_fr_front();
            break;
case 5:
            d1.delete_fr_rear();
            break;

case 6:
            exit(1);
            break;

default:
```

```
                cout<<"Invalid choice";
                break;
            }

        }while(c!=7);
        return 0;

    }
```

Output:

```
****DEQUEUE OPERATION****

1-Insert at beginning
2-Insert at end
3_Display
4_Deletion from front
5-Deletion from rear
6_Exit
Enter your choice<1-4>:1
Enter the element to be inserted:45


****DEQUEUE OPERATION****

1-Insert at beginning
2-Insert at end
3_Display
4_Deletion from front
5-Deletion from rear
6_Exit
Enter your choice<1-4>:2
Enter the element to be inserted:34


****DEQUEUE OPERATION****

1-Insert at beginning
2-Insert at end
3_Display
4_Deletion from front
5-Deletion from rear
6_Exit
Enter your choice<1-4>:3
45 34
```



Result : The Dequeue executed successfully.

Program 13: 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.

```
#include<iostream>
#include<windows.h>
using namespace std;
const int MAX=5;

class PizzaParlour
{
    int front,rear;
    int orders[MAX];
    public:
        PizzaParlour()
        {
            front=rear=-1;
        }
        bool addOrder(int data);
        void serveOrder();
        void display();
};
bool PizzaParlour::addOrder(int id){
    if(rear==-1)
    {
        front=rear=0;
        orders[rear]=id;
        return true;
     }
```

```
        else
        {
                int pos=(rear+1)%MAX;
                if(pos==front)
                {
                        cout<<"\nCafe is Full.Please wait.\n";
                        return false;
                }
                else
                {
                        rear=pos;
                        orders[rear]=id;
                        return true;
                }
        }
}

void PizzaParlour::serveOrder()
{
        if(front==-1)
        {
                cout<<"\n No Orders in Cafe.[Cafe is Empty)\n";
                return;
        }
        else
        {
                cout<<"\n Order No. "<<orders[front]<<" is processed.\n";
                if(front==rear) //only one order
```

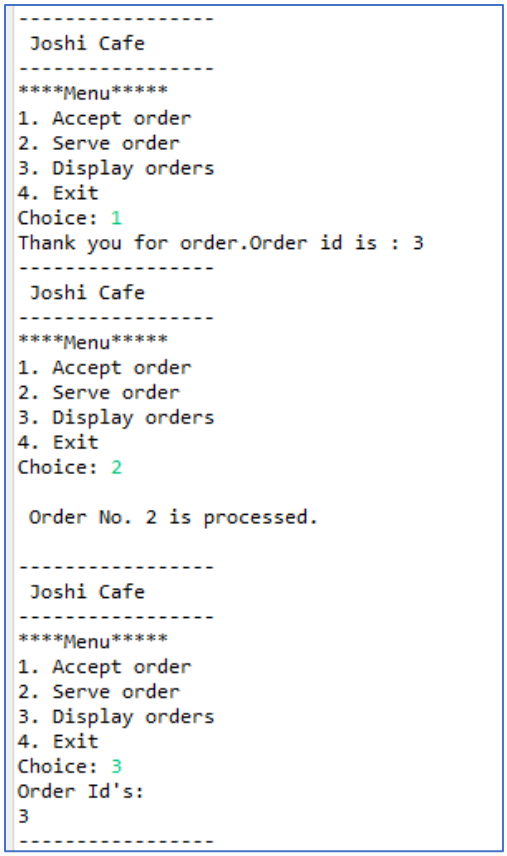
```
                {
                        front=rear=-1;
                }
                else
                {
                        front=(front+1)%MAX;
                }
        }
}

void PizzaParlour::display()
{
        int i=0;
        if(front==-1)
        {
                cout<<"\nCafe is Empty.No orders.\n";
                return;
        }
        else
        {
                cout<<"Order Id's: \n";
                for(i=front;i!=rear;i=((i+1)%MAX))
                {
                        cout<<orders[i]<<"  ";
                }
                cout<<orders[rear];
        }
}
```

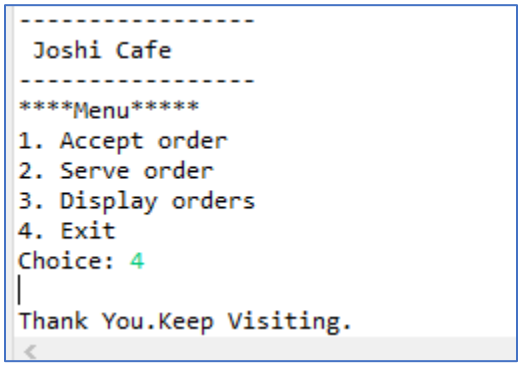
```
void intro()
{	char name[50]={"\n Joshi Cafe \n"};
		for(int i=0;name[i]!='\0';i++)
	{
		Sleep(50);
		cout<<name[i];

	}
}
int main()
{
	int ch,id=0;

	PizzaParlour q;

	do
	{
		cout<<"\n-----------------";
	intro();
		cout<<"-----------------";
		cout<<"\n****Menu*****\n";
		cout<<"1. Accept order\n";
		cout<<"2. Serve order\n";
		cout<<"3. Display orders\n";
		cout<<"4. Exit";

		cout<<"\nChoice: ";
		cin>>ch;
```

```
                switch(ch)
                {
                case 1:
                                id++;
                                if(q.addOrder(id))
                                {
                                        cout<<"Thank you for order.Order id
is : "<<id;
                                }
                                else
                                {
                                        id--;
                                }
                                break;

                case 2: q.serveOrder();
                                break;

                case 3: q.display();
                                break;
                }
        }while(ch!=4);
        cout<<"\nThank You.Keep Visiting.";

}
```

Output:

```
-----------------
 Joshi Cafe
-----------------
****Menu*****
1. Accept order
2. Serve order
3. Display orders
4. Exit
Choice: 1
Thank you for order.Order id is : 1
-----------------
 Joshi Cafe
-----------------
****Menu*****
1. Accept order
2. Serve order
3. Display orders
4. Exit
Choice: 2

 Order No. 1 is processed.

-----------------
 Joshi Cafe
-----------------
****Menu*****
1. Accept order
2. Serve order
3. Display orders
4. Exit
Choice: 1
Thank you for order.Order id is : 2
-----------------
```





Result : The Circular Queue executed successfully.