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:

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

def absentSC(listOfStudent, numberOfStudent):

count = 0

for i in range(numberOfStudent):

if listOfStudent[i]==0:

count +=1

return count

# Function for Maximum-------------------------------------------->>

def maxMarks(listOfStudent, numberOfStudent):

max = 0

for i in range(numberOfStudent):

if max<listOfStudent[i]:

max = listOfStudent[i]

return max

# Function to calculate largest marks frequency

def largestMFreq(numberOfStudent, listOfStudent):

count = 0

check = maxMarks(listOfStudent, numberOfStudent)

for i in range(numberOfStudent):

if check == listOfStudent[i]:

count += 1

return count

# Function to calculate minimum marks frequency

def minMFreq(numberOfStudent, listOfStudent):

count = 0

check = minMarks(listOfStudent, numberOfStudent)

for i in range(numberOfStudent):

if check == listOfStudent[i]:

count += 1

return count

# Function for Minimum------------------------------------------->>>

def minMarks(listOfStudent, numberOfStudent):

min = listOfStudent[0]

for i in range(numberOfStudent):

if min>listOfStudent[i]:

min = listOfStudent[i]

return min

# Function for Average--------------------------------->>>

def averageOfMarks(listOfStudent, numberOfStudent):

sumInitialize = 0

for i in range(numberOfStudent):

sumInitialize += listOfStudent[i]

return (sumInitialize/numberOfStudent)

# main program---->>

loop = True

listOfStudent = []

numberOfStudent = int(input("Enter No of Student: "))

count = 1

for i in range(numberOfStudent):

marks = int(input(f"Enter marks for student {count}: "))

listOfStudent.append(marks)

count+=1

def showList():

print("--------------------Select Your Choice From Following list-----------------\n1) Enter 1 for Average\n2) Enter 2 for Maximum\n3) Enter 3 for Minimum\n4)Enter 4 for Largest Marks Frequency\n5)Enter 5 for minimum Marks Frequency\n6)Enter 6 for Count of absent student\n7) Enter 7 to create new list of marks\n8) Enter 8 to exit")

showList()

while loop:

choice = input("Enter your choice: ")

if choice == "1":

print("Average marks obtained by student is:",averageOfMarks(listOfStudent, numberOfStudent))

elif choice == "2":

print("Maximum marks obtained by student is: ",maxMarks(listOfStudent, numberOfStudent))

elif choice == "3":

print("Minimum marks obtained by student is: ",minMarks(listOfStudent, numberOfStudent))

elif choice == "4":

print(f"{maxMarks(listOfStudent, numberOfStudent)} is largest marks and its frequency is: ",largestMFreq(numberOfStudent, listOfStudent))

elif choice == "5":

print(minMarks(listOfStudent, numberOfStudent), " is minimum marks and its frequency is: ",minMFreq(numberOfStudent, listOfStudent))

elif choice == "6":

print("Number of absent student are: ", absentSC(listOfStudent, numberOfStudent))

elif choice == "7":

print("New list created")

listOfStudent = []

numberOfStudent = int(input("Enter No of Student: "))

count = 1

for i in range(numberOfStudent):

marks = int(input(f"Enter marks for student {count}: "))

listOfStudent.append(marks)

count+=1

print("--------------------Select Your Choice From Following list-----------------\n1) Enter 1 for Average\n2) Enter 2 for Maximum\n3) Enter 3 for Minimum\n4)Enter 4 for Largest Marks Frequency\n5)Enter 5 for minimum Marks Frequency\n6)Enter 6 for Count of absent student\n7) Enter 7 to create new list of marks\n8) Enter 8 to exit")

elif choice == "8":

loop = False

else:

print("You entered wrong choice try again")

2 In second year computer engineering class, group A student’s play cricket, group B students play badminton and group C students play football. Write a Python program using functions to compute following: -

a) List of students who play both cricket and badminton

b) List of students who play either cricket or badminton but not both

c) Number of students who play neither cricket nor badminton

d) Number of students who play cricket and football but not badminton. (Note- While realizing the group, duplicate entries should be avoided, Do not use SET built-in functions

def removeDuplicate(d):

lst=[]

for i in d:

if i not in lst:

lst.append(i)

return lst

#<---------------------------------------------------------------------------------------->

# Function for finding intersection between two sets (A&B)

def intersection(lst1,lst2):

lst3=[]

for val in lst1:

if val in lst2:

lst3.append(val)

return lst3

#<------------------------------------------------------------------------------------------>

# Function for finding union of two sets (A|B)

def union(lst1,lst2):

lst3=lst1.copy()

for val in lst2:

if val not in lst3:

lst3.append(val)

return lst3

#<------------------------------------------------------------------------------------------->

# Function for finding difference between two sets (A-B)

def diff(lst1,lst2):

lst3=[]

for val in lst1:

if val not in lst2:

lst3.append(val)

return lst3

#<---------------------------------------------------------------------------------------------->

# Function for finding symmetric difference of two sets (A^B)

def sym\_diff(lst1,lst2):

lst3=[]

D1=diff(lst1,lst2)

print("Difference between Cricket and Badminton (C-B) is : ", D1)

D2=diff(lst2,lst1)

print("Difference between Badminton and Cricket (B-C) is : ", D2)

lst3=union(D1,D2)

return lst3

#<------------------------------------------------------------------------------------------------>

# Functon for finding List of students who play both cricket and badminton

def CB(lst1,lst2):

lst3=intersection(lst1,lst2)

print("\n\nList of students who play both cricket and badminton is : ", lst3)

return len(lst3)

#<------------------------------------------------------------------------------------------------>

# Function for finding List of students who play either cricket or badminton but not both

def eCeB(lst1,lst2):

lst3=sym\_diff(lst1,lst2)

print("\nList of students who play either cricket or badminton but not both is : ",lst3)

return len(lst3)

#<-------------------------------------------------------------------------------------------------->

# Function for finding Number of students who play neither cricket nor badminton

def nCnB(lst1,lst2,lst3):

lst4=diff(lst1,union(lst2,lst3))

print("\n\nList of students who play neither cricket nor badminton is : ",lst4)

return len(lst4)

#<--------------------------------------------------------------------------------------------------->

# Function for finding Number of students who play cricket and football but not badminton

def CBnF(lst1,lst2,lst3):

lst4=diff(intersection(lst1,lst2),lst3)

print("\n\nList of students who play cricket and football but not badminton is : ",lst4)

return len(lst4)

#<----------------------------------------------------------------------------------------------------->

# Main function

# Creating an empty list for SE COMP

SEComp = []

n = int(input("\nEnter number of students in SE COMP: "))

print("Enter the names of",n,"students (Please press ENTER after entering each students name) :")

for i in range(0, n):

ele = input()

SEComp.append(ele) # adding the element

print("Original list of students in SEComp : " + str(SEComp))

#<------------------------------------------------------------------------------------------------------->

# Creating an empty list for Cricket

Cricket = []

n = int(input("\n\nEnter number of students who play cricket : "))

print("Enter the names of",n,"students who play cricket (Please press ENTER after entering each students name) :")

for i in range(0, n):

ele = input()

Cricket.append(ele) # adding the element

print("Original list of students playing cricket is :" +str(Cricket))

Cricket=removeDuplicate(Cricket)

print("The list of students playing cricket after removing duplicates : " +str(Cricket))

#<------------------------------------------------------------------------------------------------------->

# Creating an empty list for Football

Football = []

n = int(input("\n\nEnter number of students who play football : "))

print("Enter the name of",n,"students who play football (Please press ENTER after entering each students name) :")

for i in range(0, n):

ele = input()

Football.append(ele) # adding the element

print("Original list of students playing football :" +str(Football))

Football=removeDuplicate(Football)

print("The list of students playing football after removing duplicates : " +str(Football))

#<-------------------------------------------------------------------------------------------------------->

# Creating an empty list for Badminton

Badminton = []

n = int(input("\n\nEnter number of students who play badminton : "))

print("Enter the name of",n,"students who play badminton (Please press ENTER after entering each students name) :")

for i in range(0, n):

ele = input()

Badminton.append(ele) # adding the element

print("Original list of students playing badminton :" +str(Badminton))

Badminton=removeDuplicate(Badminton)

print("The list of students playing badminton after removing duplicates : " +str(Badminton))

#<---------------------------------------------------------------------------------------------------------->

flag=1

while flag==1:

print("\n\n--------------------MENU--------------------\n")

print("1. List of students who play both cricket and badminton")

print("2. List of students who play either cricket or badminton but not both")

print("3. List of students who play neither cricket nor badminton")

print("4. Number of students who play cricket and football but not badminton")

print("5. Exit\n")

ch=int(input("Enter your Choice (from 1 to 5) :"))

if ch==1:

print("Number of students who play both cricket and badminton : ", CB(Cricket,Badminton))

a = input("\n\nDo you want to continue (yes/no) :")

if a == "yes":

flag = 1

else:

flag = 0

print("Thanks for using this program!")

elif ch==2:

print("Number of students who play either cricket or badminton but not both : ", eCeB(Cricket, Badminton))

a = input("\n\nDo you want to continue (yes/no) :")

if a == "yes":

flag = 1

else:

flag = 0

print("Thanks for using this program!")

elif ch==3:

print("Number of students who play neither cricket nor badminton : ", nCnB(SEComp,Cricket,Badminton))

a = input("\n\nDo you want to continue (yes/no) :")

if a == "yes":

flag = 1

else:

flag = 0

print("Thanks for using this program!")

elif ch==4:

print("Number of students who play cricket and football but not badminton : ", CBnF(Cricket,Football,Badminton))

a = input("\n\nDo you want to continue (yes/no) :")

if a == "yes":

flag = 1

else:

flag = 0

print("Thanks for using this program!")

elif ch==5:

flag=0

print("Thanks for using this program!")

else:

print("!!Wrong Choice!! ")

a=input("\n\nDo you want to continue (yes/no) :")

if a=="yes":

flag=1

else:

flag=0

print("Thanks for using this program!")

3.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

import numpy

# initializing matrices

x = numpy.array([[1, 2], [4, 5]])

y = numpy.array([[7, 8], [9, 10]])

# using add() to add matrices

print("The element wise addition of matrix is : ")

print(numpy.add(x, y))

# using subtract() to subtract matrices

print("The element wise subtraction of matrix is : ")

print(numpy.subtract(x, y))

# using dot() to multiply matrices

print ("The product of matrices is : ")

print (numpy.dot(x,y))

# using "T" to transpose the matrix

print("The transpose of given matrix is : ")

print(x.T)

4.Write a python program to maintain club members, sort on roll numbers in ascending order. Write function “Ternary\_Search” to search whether particular student is member of club or not. Ternary search is modified binary search that divides array into 3 halves instead of two.

def accept\_roll():

roll\_no = []

no\_of\_students = int(input("Enter the number of students : "))

for i in range(no\_of\_students):

roll\_no.append(int(input("Enter Roll Number of Student {0} : ".format(i+1))))

return roll\_no

#<--------------------------------------------------------------------------------------------->

# Printing the Roll Numbers of the Students

def print\_roll(roll\_no):

for i in range(len(roll\_no)):

print(roll\_no[i],sep = "\n")

#<--------------------------------------------------------------------------------------------->

# Insertion Sort for Sorting the list of Roll Numbers

def insertion\_sort(roll\_no):

for i in range(1,len(roll\_no)):

key = roll\_no[i]

j = i-1;

while j >= 0 and key < roll\_no[j]:

roll\_no[j+1] = roll\_no[j]

j -= 1

roll\_no[j+1] = key

return roll\_no

#<---------------------------------------------------------------------------------------------->

# Function for performing Non-Recursive Ternary Search

def NR\_Ternary\_Search(roll,roll\_find):

left = 0

right = len(roll) - 1

while left <= right:

mid1 = left + (right - left) // 3

mid2 = left + 2 \* (right - left) // 3

if roll\_find == roll[left]:

return left

elif roll\_find == roll[right]:

return right

elif roll\_find < roll[left] or roll\_find > roll[right]:

return -1

elif roll\_find <= roll[mid1]:

right = mid1

elif roll\_find > roll[mid1] and roll\_find <= roll[mid2]:

left = mid1 + 1

right = mid2

else:

left = mid2 + 1

return -1

#<------------------------------------------------------------------------------------------------->

# Function for performing Recursive Ternary Search

def R\_Ternary\_Search(roll, left, right, roll\_find):

if (right >= left):

mid1 = left + (right - left) // 3

mid2 = right - (right - left) // 3

if (roll[mid1] == roll\_find):

return mid1

if (roll[mid2] == roll\_find):

return mid2

if (roll\_find < roll[mid1]):

return R\_Ternary\_Search(roll, left, mid1 - 1, roll\_find)

elif (roll\_find > roll[mid2]):

return R\_Ternary\_Search(roll, mid2 + 1, right, roll\_find)

else:

return R\_Ternary\_Search(roll, mid1 + 1, mid2 - 1, roll\_find)

return -1

#<---------------------------------------------------------------------------------------------------->

# Main

unsort\_Roll = []

sort\_Roll = []

flag = 1

while flag == 1:

print("\n---------------------MENU---------------------")

print("1. Accept Student Roll Numbers")

print("2. Display the Roll Numbers of Student")

print("3. Sort Roll Numbers from the list")

print("4. Perform Non-Recursive Ternary Search")

print("5. Perform Recursive Ternary Search")

print("6. Exit\n")

ch = int(input("Enter your choice (from 1 to 6) : "))

if ch == 1:

unsort\_Roll = accept\_roll()

elif ch == 2:

print\_roll(unsort\_Roll)

elif ch == 3:

print("Elements after performing Insertion Sort : \n")

sort\_Roll = insertion\_sort(unsort\_Roll)

print\_roll(sort\_Roll)

elif ch == 4:

find\_roll = int(input("Enter the Roll Number to be searched : "))

index = NR\_Ternary\_Search(sort\_Roll,find\_roll)

if index != -1:

print("The Roll Number",find\_roll,"is found at position",index+1)

else:

print("Roll Number",find\_roll,"nor found!!")

elif ch == 5:

find\_roll = int(input("Enter the Roll Number to be searched : "))

left = 0

right = len(sort\_Roll) - 1

index = R\_Ternary\_Search(sort\_Roll,left,right,find\_roll)

if index != -1:

print("The Roll Number",find\_roll,"is found at position",index+1)

else:

print("Roll Number",find\_roll,"nor found!!")

elif ch == 6:

print("Thanks for using this program!!")

flag=0

else:

print("Wrong choice!!")

        flag = 0

6.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

def countingSort(arr, exp1):

n = len(arr)

# The output array elements that will have sorted arr

output = [0] \* (n)

# initialize count array as 0

count = [0] \* (10)

# Store count of occurrences in count[]

for i in range(0, n):

index = (arr[i]/exp1)

count[int((index)%10)] += 1

# Change count[i] so that count[i] now contains actual

# position of this digit in output array

for i in range(1,10):

count[i] += count[i-1]

# Build the output array

i = n-1

while i>=0:

index = (arr[i]/exp1)

output[ count[ int((index)%10) ] - 1] = arr[i]

count[int((index)%10)] -= 1

i -= 1

# Copying the output array to arr[],

# so that arr now contains sorted numbers

i = 0

for i in range(0,len(arr)):

arr[i] = output[i]

# Method to do Radix Sort

def radixSort(arr):

# Find the maximum number to know number of digits

max1 = max(arr)

# Do counting sort for every digit. Note that instead

# of passing digit number, exp is passed. exp is 10^i

# where i is current digit number

exp = 1

while max1/exp > 0:

countingSort(arr,exp)

exp \*= 10

# Driver code to test above

arr = [ 170, 45, 75, 90, 802, 24, 2,66]

radixSort(arr)

for i in range(len(arr)):

print(arr[i])

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

a) Selection Sort

b) Bubble sort and display top five scores

def Selection\_Sort(marks):

for i in range(len(marks)):

# Find the minimum element in remaining unsorted array

min\_idx = i

for j in range(i + 1, len(marks)):

if marks[min\_idx] > marks[j]:

min\_idx = j

# Swap the minimum element with the first element

marks[i], marks[min\_idx] = marks[min\_idx], marks[i]

print("Marks of students after performing Selection Sort on the list : ")

for i in range(len(marks)):

print(marks[i])

#<--------------------------------------------------------------------------------------->

# Function for Bubble Sort of elements

def Bubble\_Sort(marks):

n = len(marks)

# Traverse through all array elements

for i in range(n - 1):

# Last i elements are already in place

for j in range(0, n - i - 1):

# Traverse the array from 0 to n-i-1

# Swap if the element found is greater than the next element

if marks[j] > marks[j + 1]:

marks[j], marks[j + 1] = marks[j + 1], marks[j]

print("Marks of students after performing Bubble Sort on the list :")

for i in range(len(marks)):

print(marks[i])

#<--------------------------------------------------------------------------------------->

# Function for displaying top five marks

def top\_five\_marks(marks):

print("Top",len(marks),"Marks are : ")

print(\*marks[::-1], sep="\n")

#<---------------------------------------------------------------------------------------->

# Main

marks=[]

n = int(input("Enter number of students whose marks are to be displayed : "))

print("Enter marks for",n,"students (Press ENTER after every students marks): ")

for i in range(0, n):

ele = int(input())

marks.append(ele) # adding the element

print("The marks of",n,"students are : ")

print(marks)

flag=1;

while flag==1:

print("\n---------------MENU---------------")

print("1. Selection Sort of the marks")

print("2. Bubble Sort of the marks")

print("3. Exit")

ch=int(input("\n\nEnter your choice (from 1 to 3) : "))

if ch==1:

Selection\_Sort(marks)

a=input("\nDo you want to display top marks from the list (yes/no) : ")

if a=='yes':

top\_five\_marks(marks)

else:

print("\nThanks for using this program!")

flag=0

elif ch==2:

Bubble\_Sort(marks)

a = input("\nDo you want to display top five marks from the list (yes/no) : ")

if a == 'yes':

top\_five\_marks(marks)

else:

print("\nThanks for using this program!")

flag = 0

elif ch==3:

print("\nThanks for using this program!!")

flag=0

else:

print("\nEnter a valid choice!!")

print("\nThanks for using this program!!")

        flag=0

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) Display list in reverse order using recursion

e) 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;

 }

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 display

a) 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

{ int roll;

struct node \*next;

};

class info

{ node \*head1=NULL,\*temp1=NULL,\*head2=NULL,\*temp2=NULL,\*head=NULL,\*temp=NULL,\*h1=NULL,\*head3=NULL,\*temp3=NULL;

int c,i,f,j,k;

public:

node \*create();

void insert();

void allstud();

void vanila();

void butters();

void uice();

void nice();

void notice();

void ovanila();

void obutters();

void display();

} ;

node \*info::create()

{ node \*p=new(struct node);

cout<<"enter student rollno";

cin>>c;

p->roll=c;

p->next=NULL;

return p;

}

void info::insert()

{

node \*p=create();

if(head==NULL)

{ head=p;

}

else

{ temp=head;

while(temp->next!=NULL)

{ temp=temp->next; }

temp->next=p;

}

}

void info::display()

{ temp=head;

while(temp->next!=NULL)

{ cout<<"\n"<<temp->roll;

temp=temp->next;

} cout<<"\n"<<temp->roll;

}

void info::allstud()

{cout<<"enter no. of students";

cin>>k;

head=NULL;

for(i=0;i<k;i++)

{ insert();

h1=head;

} display();

head=NULL;

}

void info::vanila()

{

cout<<"enter no. of students who like vanila";

cin>>k;

head=NULL;

for(i=0;i<k;i++)

{ insert();

head1=head;

} display();

head=NULL;

}

void info::butters()

{

cout<<"enter no. of students who like butterscotch";

cin>>j;

for(i=0;i<j;i++)

{ insert();

head2=head;

} display();

head=NULL;

}

void info::uice()

{ cout<<"students who like vanila or butterscotch\n";

temp1=head1;

while(temp1!=NULL)

{

node \*p=new(struct node);

p->roll=temp1->roll;

p->next=NULL;

if(head3==NULL)

{ head3=p;

}

else

{ temp3=head3;

while(temp3->next!=NULL)

{ temp3=temp3->next; }

temp3->next=p;

}

temp1=temp1->next;

}

temp2=head2;

while(temp2!=NULL)

{ f=0;

temp1=head1;

while(temp1!=NULL)

{

if(temp2->roll==temp1->roll)

{ f=1; }

temp1=temp1->next;

}

if(f==0)

{

node \*p=new(struct node);

p->roll=temp2->roll;

p->next=NULL;

if(head3==NULL)

{ head3=p;

}

else

{ temp3=head3;

while(temp3->next!=NULL)

{ temp3=temp3->next; }

temp3->next=p;

}

}

temp2=temp2->next;

}

temp3=head3;

while(temp3->next!=NULL)

{ cout<<"\n"<<temp3->roll;

temp3=temp3->next;

} cout<<"\n"<<temp3->roll;

}

void info::ovanila()

{

cout<<"\nstudents like only vanila \n";

temp1=head1;

while(temp1!=NULL)

{ temp2=head2;

f=0;

while(temp2!=NULL)

{ if(temp1->roll==temp2->roll)

{ f=1; }

temp2=temp2->next;

}

if(f==0)

{ cout<<"\n"<<temp1->roll; }

temp1=temp1->next;

}

}

void info::obutters()

{

cout<<"\nstudents like only butterscotch\n";

temp2=head2;

while(temp2!=NULL)

{ temp1=head1;

f=0;

while(temp1!=NULL)

{ if(temp2->roll==temp1->roll)

{ f=1; }

temp1=temp1->next;

}

if(f==0)

{ cout<<"\n"<<temp2->roll; }

temp2=temp2->next;

}

}

void info::nice()

{

cout<<"\nstudents who like both vanila and butterscotch\n";

temp1=head1;

while(temp1!=NULL)

{ temp2=head2;

while(temp2!=NULL)

{ if(temp1->roll==temp2->roll)

{ cout<<"\n"<<temp1->roll; }

temp2=temp2->next;

}

temp1=temp1->next;

}

}

void info::notice()

{

cout<<"\nstudents who like neither vanila nor butterscotch\n";

temp=h1;

while(temp!=NULL)

{ temp3=head3;

f=0;

while(temp3!=NULL)

{ if(temp->roll==temp3->roll)

{ f=1; }

temp3=temp3->next;

}

if(f==0)

{ cout<<"\n"<<temp->roll; }

temp=temp->next;

}

}

int main()

{ info s;

int i;

char ch;

do{

cout<<"\n choice the options";

cout<<"\n 1. To enter all students rollno ";

cout<<"\n 2. To enter the rollno of student who like vanila";

cout<<"\n 3. To enter the rollno of student who like butterscotch";

cout<<"\n 4. To display the rollno of student who like vanila or butterscotch";

cout<<"\n 5. To display the rollno of student who like only vanila";

cout<<"\n 6. To display the rollno of student who like only butterscotch";

cout<<"\n 7. To display the rollno of student who like both vanila and butterscotch ";

cout<<"\n 8. To display the rollno of student who neither like vanila nor butterscotch";

cin>>i;

switch(i)

{ case 1: s.allstud();

break;

case 2: s.vanila();

break;

case 3: s.butters();

break;

case 4: s.uice();

break;

case 5: s.ovanila();

break;

case 6: s. obutters();

break;

case 7: s.nice();

break;

case 8: s.notice();

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;

}

8. 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.

#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 SeatStatus\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;

}

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 functions

a) To print original string followed by reversed string using stack

b) To check whether given string is palindrome or not #include

#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';

cout<<endl<<c<<" is pushed on stack ...";

}

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();

}

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. ii. ii. Input Postfix expression must be in a desired format.
3. iii. iii. Only '+', '-', '\*' and '/ ' operators are expected.

#include<iostream>

#include<cctype>

#include<stack>

using namespace std;

// returns the value when a specific operator

// operates on two operands

int eval(int op1, int op2, char operate) {

switch (operate) {

case '\*': return op2 \* op1;

case '/': return op2 / op1;

case '+': return op2 + op1;

case '-': return op2 - op1;

default : return 0;

}

}

int getWeight(char ch) {

switch (ch) {

case '/':

case '\*': return 2;

case '+':

case '-': return 1;

default : return 0;

}

}

// evaluates the postfix operation

// this module neither supports multiple digit integers

// nor looks for valid expression

// However it can be easily modified and some additional

// code can be added to overcome the above mentioned limitations

// it's a simple function which implements the basic logic to

// evaluate postfix operations using stack

int evalPostfix(char postfix[], int size) {

stack<int> s;

int i = 0;

char ch;

int val;

while (i < size) {

ch = postfix[i];

if (isdigit(ch)) {

// we saw an operand

// push the digit onto stack

s.push(ch-'0');

}

else {

// we saw an operator

// pop off the top two operands from the

// stack and evalute them using the current

// operator

int op1 = s.top();

s.pop();

int op2 = s.top();

s.pop();

val = eval(op1, op2, ch);

// push the value obtained after evaluating

// onto the stack

s.push(val);

}

i++;

}

return val;

}

// main

int main() {

char postfix[] = {'a','b','c','+','\*','d','/'};

int i=0;

// cout<<"Postfix 0"<<postfix[i];

while(postfix[i]!='\0')

{

if(getWeight(postfix[i])==0)

{

cout<<" Enter value for "<<postfix[i]<<" = ";

cin>>postfix[i];

}

i++;

}

int size = sizeof(postfix);

int val = evalPostfix(postfix, size);

cout<<"\nExpression evaluates to "<<val;

cout<<endl;

return 0;

}

InfixToPostfix.cpp

#include<iostream>

#include<cstring>

#include<stack>

using namespace std;

// get weight of operators as per precedence

// higher weight given to operators with higher precedence

// for non operators, return 0

int getWeight(char ch) {

switch (ch) {

case '/':

case '\*': return 2;

case '+':

case '-': return 1;

default : return 0;

}

}

// convert infix expression to postfix using a stack

void infix2postfix(char infix[], char postfix[], int size) {

stack<char> s;

int weight;

int i = 0;

int k = 0;

char ch;

// iterate over the infix expression

while (i < size) {

ch = infix[i];

if (ch == '(') {

// simply push the opening parenthesis

s.push(ch);

i++;

continue;

}

if (ch == ')') {

// if we see a closing parenthesis,

// pop of all the elements and append it to

// the postfix expression till we encounter

// a opening parenthesis

while (!s.empty() && s.top() != '(') {

postfix[k++] = s.top();

s.pop();

}

// pop off the opening parenthesis also

if (!s.empty()) {

s.pop();

}

i++;

continue;

}

weight = getWeight(ch);

if (weight == 0) {

// we saw an operand

// simply append it to postfix expression

postfix[k++] = ch;

}

else {

// we saw an operator

if (s.empty()) {

// simply push the operator onto stack if

// stack is empty

s.push(ch);

}

else {

// pop of all the operators from the stack and

// append it to the postfix expression till we

// see an operator with a lower precedence that

// the current operator

while (!s.empty() && s.top() != '(' &&

weight <= getWeight(s.top())) {

postfix[k++] = s.top();

s.pop();

}

// push the current operator onto stack

s.push(ch);

}

}

i++;

}

// pop of the remaining operators present in the stack

// and append it to postfix expression

while (!s.empty()) {

postfix[k++] = s.top();

s.pop();

}

postfix[k] = 0; // null terminate the postfix expression

}

// main

int main() {

char infix[100];//"A\*(B+C)/D";

cout<<"\nENter Infix Operation:";

cin>>infix;

int size = strlen(infix);

char postfix[size];

infix2postfix(infix,postfix,size);

cout<<"\nInfix Expression :: "<<infix;

cout<<"\nPostfix Expression :: "<<postfix;

cout<<endl;

return 0;

}

11.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.

#include <iostream>

#define MAX 10

using namespace std;

struct queue

{ int data[MAX];

int front,rear;

};

class Queue

{ struct queue q;

public:

Queue(){q.front=q.rear=-1;}

int isempty();

int isfull();

void enqueue(int);

int delqueue();

void display();

};

int Queue::isempty()

{

return(q.front==q.rear)?1:0;

}

int Queue::isfull()

{ return(q.rear==MAX-1)?1:0;}

void Queue::enqueue(int x)

{q.data[++q.rear]=x;}

int Queue::delqueue()

{return q.data[++q.front];}

void Queue::display()

{ int i;

cout<<"\n";

for(i=q.front+1;i<=q.rear;i++)

cout<<q.data[i]<<" ";

}

int main()

{ Queue obj;

int ch,x;

do{ cout<<"\n 1.Insert Job\n 2.Delete Job\n 3.Display\n 4.Exit\n Enter your choice : ";

cin>>ch;

switch(ch)

{ case 1: if (!obj.isfull())

{ cout<<"\n Enter data : \n";

cin>>x;

obj.enqueue(x);

cout<<endl;

}

else

cout<< "Queue is overflow!!!\n\n";

break;

case 2: if(!obj.isempty())

cout<<"\n Deleted Element = "<<obj.delqueue()<<endl;

else

{ cout<<"\n Queue is underflow!!!\n\n"; }

cout<<"\nRemaining Jobs : \n";

obj.display();

break;

case 3: if (!obj.isempty())

{ cout<<"\n Queue contains : \n";

obj.display();

}

else

cout<<"\n Queue is empty!!!\n\n";

break;

case 4: cout<<"\n Exiting Program.....";

}

}while(ch!=4);

return 0;

}

13.Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed can not be cancelled. Write C++ program to simulate the system using circular queue using array

#include<iostream>

using namespace std;

struct piz

{ int ono,ot;

};

class queue

{ piz p[10];

int count,c,rear,front,f;

public:

queue()

{ c=0; f=0; count=0; front=0; rear=-1;}

void insert();

void display();

void del();

};

void queue::insert()

{ int i=0,n;

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\n|1. corn pizza";

cout<<"\n|2. pizza\t\t|";

cout<<"\n|3. chicken pizza\t|";

cout<<"\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\n---------------------------------------------------------------------";

cout<<"\n how many order you want to place";

cout<<"\n---------------------------------------------------------------------";

cin>>n;

for(i=0;i<n;i++)

{ if(count>=10)

{ cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\n|over flow|";

cout<<"\n|\_\_\_\_\_\_\_\_\_\_|"; break;

}

else

{

c++;

if(rear==9)

{ rear=0; }

else

{ rear++; }

cout<<"\nenter your order";

cin>>p[rear].ot;

p[rear].ono=c; count++;

}

}

cout<<"\n-------------------------------------------------------------------";

}

void queue::display()

{ int i=0,n,f=front;

cout<<"\n-------------------------------------------------------------------";

cout<<"\n how many order you want to display";

cout<<"\n-------------------------------------------------------------------";

cin>>n;

for(i=0;i<n;i++)

{ if(i>=count)

{ cout<<"\nall data dispalyed"; break; }

else

{

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\nreceive your order please ,no. ->"<<p[f].ono;

cout<<"\n which is:";

if(p[f].ot==1)

{ cout<<" corn pizza"; }

if(p[f].ot==2)

{ cout<<" pizza"; }

if(p[f].ot==3)

{ cout<<" chicken pizza"; }

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

f++;

if(f==10)

{ f=0; }

}

} cout<<"\n-------------------------------------------------------------------";

}

void queue::del()

{ int i=0,n;

cout<<"\n-------------------------------------------------------------------";

cout<<"\n how many order you want to delete";

cout<<"\n-------------------------------------------------------------------";

cin>>n;

for(i=0;i<n;i++)

{ if(count==0)

{ cout<<"\nunderflow"; break; }

else

{

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\ndeleted order no."<<p[front].ono;

cout<<"\n which is:";

if(p[front].ot==1)

{ cout<<" corn pizza "; }

if(p[front].ot==2)

{ cout<<" pizza "; }

if(p[front].ot==3)

{ cout<<" chicken pizza"; }

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

p[front].ono=0; p[front].ot=0;

front++;

if(front==10)

{ front=0; } count--;

}

}

cout<<"\n-------------------------------------------------------------------";

}

int main()

{ queue q; int y=1,x;

while(y==1)

{ cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\n|enter the choice ";

cout<<"\n|1.to place order ";

cout<<"\n|2.to display order|";

cout<<"\n|3.to delete order |";

cout<<"\n|4.to exit ";

cout<<"\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cin>>x;

switch(x)

{ case 1: q.insert();

break;

case 2: q.display();

break;

case 3: q.del();

break;

case 4: y=0;

break;

default: cout<<"\n unknown entry";

}

}

    return 0;

}

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

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;

}