**Roll No: 09 (SE-A) DSL**

**'''Practicle No. 1 :** In a second year computer engineering class, group A students 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.'''

#function for removing duplicate entries from the group

def removeDuplicate(d):

    lst=[]

    for i in d:

        if i not in lst:

            lst.append(i)

    return lst

#function for finding the intersection betn 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 vqal not in lst2:

            lst3.append(val)

    return lst3

#finding for finding symmetric difference to two sets(A^ b)

def sym\_diff(lst1, lst2):

    lst3 = []

    D1 = diff(lst1, lst2)

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

    D1 = diff(lst2, lst1)

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

    lst3=union(D1,D2)

    return lst3

#function for finding list of students who play both cricket & badminton

def CB(lst1, lst2):

    lst3 = intersection(lst1,lst2)

    print("\n\n list of students who play both cricket & Badminton len is:", lst3)

    return len (lst3)

#function for finding no.of students who play neither cricket nor badminton

def nCnB(lst1,lst2,lst3):

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

    print("\n\n list of students who play neighter cricket nor badminton is:", lst4)

    return(len(lst4))

#function for finding no.of students who play cricket, football but not badminton

def nFnB(lst1,lst2,lst3):

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

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

    return(len(lst4))

#main function

#creating an empty list for SEcomp students

SEcomp = []

n = int(input("\n Enter no. of student in SEcomp:"))

print("Enter the names of", n,"students:")

for i in range(0,n):

    ele = input()

    SEcomp.append(ele)

    print("Original list of student in SEcomp:"+str(SEcomp))

#creating an empty list for  cricket students

cricket = []

n = int(input("\n Enter no. of student who play cricket:"))

print("Enter the names of", n,"students:")

for i in range(0,n):

    ele = input()

    cricket.append(ele)

print("Original list of student 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 students

football = []

n = int(input("\n Enter no. of student who play football:"))

print("Enter the names of", n,"students:")

for i in range(0,n):

    ele = input()

    football.append(ele)

print("Original list of student playing football is:"+str(football))

football = removeDuplicate(football)

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

#creating an empty list for Badminton students

badminton = []

n = int(input("\n Enter no. of student who play badminton:"))

print("Enter the names of", n,"students:")

for i in range(0,n):

    ele = input()

    badminton.append(ele)

print("Original list of student playing badminton is:"+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 & football but not badminton:")

    print("5.Exit\n")

    break

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

if ch == 1:

    print("No.of students who play both cricket & 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("No.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("No.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("No.of students who play cricket and football but not badminton:", CFnB(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

    football.append(ele)

    print("Thanks for using this program!")

-------------------OUTPUT-------------------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a1.py

 Enter no. of student in SEcomp:2

Enter the names of 2 students:

maya payal

Original list of student in SEcomp:['maya payal']

Original list of student in SEcomp:['maya payal', '']

 Enter no. of student who play cricket:3

Enter the names of 3 students:

a b c

Original list of student playing cricket is:['a b c', '', '']

The list of students playing cricket after removing duplicates:['a b c', '']

 Enter no. of student who play football:1

Enter the names of 1 students:

jahn

Original list of student playing football is:['jahn']

The list of students playing badminton after removing duplicates:['jahn']

 Enter no. of student who play badminton:2

Enter the names of 2 students:

chitra sneha gauri

Original list of student playing badminton is:['chitra sneha gauri', '']

The list of students playing badminton after removing duplicates:['chitra sneha gauri', '']

\_\_\_\_\_menu\_\_\_\_\_

1.List of students who play both cricket and badminton:

2.List of students who play either cricket or badminton but not both:

3.List of students who play neither cricket nor badminton:

4.Number of students who play cricket & football but not badminton:

5.Exit

Enter your choice(from 1 to 5):1

 list of students who play both cricket & Badminton len is: ['']

No.of students who play both cricket & badminton: 1

Do you want to continue(yes/no):yes

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ '''

**'''Practical Number 2 :** Write a python program to store marks stored in subject "Fundamentals of Data Structure" by

                         N students in the class. Write functions to compute following:

                         1. The average score of the class.

                         2. Highest score and lowest score of the class.

                         3. Count of students who were absent for the test.

                         4. Display mark with highest frequency.'''

#function for average score of the class

def average(listofmarks):

    sum=0

    count=0

    for i in range (len(listofmarks)):

      if listofmarks[i]!=-999:

        sum+=listofmarks[i]

        count+=1

      avg=sum/count

      print("total marks:,")

      print("average marks:{:.2f}".format(avg))

#function for heighest score in the test for the class

def Maximum(listofmarks):

    for i in range(len(listofmarks)):

      if listofmarks[i]!=-999:

        Max=listofmarks[0]

        break

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

      if listofmarks[i]>Max:

        Max=listofmarks[i]

      return(Max)

#function for lowest score in the test for the class

def Minimum(listofmarks):

    for i in range(len(listofmarks)):

      if listofmarks[i]!=-999:

        Min=listofmarks[0]

        break

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

      if listofmarks[i]>Min:

        Min=listofmarks[i]

      return(Min)

#function for counting the number of student absent for the test

def absentcount(listofmarks):

     count=0

     for i in range(len(listofmarks)):

      if listofmarks[i]==-999:

         count+=1

      return(count)

#function for displaying marks with highest frequency

def maxfrequency(listofmarks):

     i=0

     Max=0

     print("marks|frequency")

     for j in listofmarks:

        if(listofmarks.index(j)==i):

           print("j,   |   ",listofmarks.count(j))

           if listofmarks.count(j)>Max:

             Max=listofmarks.count(j)

             marks=j

             i=i+1

             return(marks,Max)

#main function

marksinFDS=[]

numberofstudent=int(input("enter total number of student:"))

for i in range(numberofstudent):

 marks=int(input("enter marks of student "+str(i+1)+" : "))

 marksinFDS.append(marks)

flag=1

while flag==1:

     print("\n\n   MENU   \n")

     print("1.total and average marks of the class")

     print("2.highest and lowest marks in the class")

     print("3.number of student absent for the test")

     print("4.marks with highest frequency")

     print("5.exit\n")

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

     if ch==1:

       average(marksinFDS)

       a=input("do you want to continue(yes/no):")

       if a=="yes":

           flag=1

       else:

           flag=0

           print("thanks for using this program!")

     if ch==2:

       print("highest score in class:",Maximum(marksinFDS))

       print("lowest score in class:",Minimum(marksinFDS))

       a=input("do 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 student absent for the test:",absentcount(marksinFDS))

       a=input("do you want to continue(yes/no):")

       if a=="yes":

           flag=1

       else:

           flag=0

           print("thanks for using this program!")

     elif ch==4:

       mark,fr=maxfrequency(marksinFDS)

       print("highest frequency of marks {0} that is {1}".format(marks,fr))

       a=input("do 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("do you want to continue(yes/no):")

           if a=="yes":

             flag=1

           else:

             flag=0

           print("thanks for using this program!")

  -----------------------OUTPUT---------------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a2.py

enter total number of student:3

enter marks of student 1 : 45

enter marks of student 2 : 78

enter marks of student 3 : 89

   MENU

1.total and average marks of the class

2.highest and lowest marks in the class

3.number of student absent for the test

4.marks with highest frequency

5.exit

Enter your choice(from 1 to 5):1

total marks:,

average marks:45.00

total marks:,

average marks:61.50

total marks:,

average marks:70.67

do you want to continue(yes/no):y

thanks for using this program!

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ '''

"""

**practicale No 3:** Write a python program to compute following operations on String:

a) To display word with the longest length

b) To determines the frequency of occurrence of particular character in the string

c) To check whether given string is palindrome or not

d) To display index of first appearance of the substring

e) To count the occurrences of each word in a given string

"""

def Display\_word\_with\_longest\_length() :

   Str = input("Enter the main string : ")  # Its assumed that string contain only characters and spaces (multile spaces are allowed)

   M\_str = ""

   i = 0

   while( i < len(Str)) :

      word = ""

      while(Str[i] != ' ') :

         word += Str[i]

         i = i + 1

         if( i == len(Str)) :

             break

      if(i != len(Str)) :

         while(Str[i] == ' ') :

            i = i + 1

      if(len(M\_str) < len(word)) :

         M\_str = word

   print("\tWord with longest length is %s having lenght %d\n\n"%(M\_str,len(M\_str)))

def Determine\_frequency\_of\_occurrence\_of\_particular\_character\_in\_string() :

   Str = input("Enter the string : ")

   C = input("Enter the character  : ")

   print("\tString : %s"%Str)

   print("\tCharacter : %s"%C)

   count = 0

   for i in range(len(Str)) :

      if(Str[i] == C) :

         count += 1

   print("\tFrequency of occurrence of character(%s) in string(%s) is %d\n\n"%(C,Str,count))

def Check\_for\_palindrome() :

   Str = input("Enter the string to be checked : ")

   b = 0

   e = len(Str) - 1

   while( b < e) :

      if(Str[b] != Str[e]) :

         break

      b += 1

      e -= 1

   if(b < e) :

      print("\t%s string is not an palindrome string\n\n"%Str)

   else :

      print("\t%s string is an palindrome string\n\n"%Str)

def display\_index\_of\_first\_appearance\_of\_the\_substring() :

   M = input("Enter the main string : ")

   S = input("Enter the sub string to check : ")

   print("Main String : %s"%M)

   print("Substring String : %s"%S)

   L1 = len(M)

   L2 = len(S)

   if(L1 >= L2) :

      for i in range((L1 - L2 + 1)) :

         flag = 1

         for j in range(L2):

            if(M[i+j] != S[j]) :

               flag = 0

               break

         if(flag == 1) :

            print("Substring %s found at index %d\n\n"%(S,i))

            break;

      if(flag == 0) :

         print("Substring not found in the main string\n\n")

   else :

      print("Substring is greater than main string\n\n")

def Count\_\_occurrences\_of\_each\_word\_in\_given\_string() :

   Str = input("Enter the main string : ")  # Its assumed that string contain only characters and spaces (multile spaces are allowed)

   i = 0

   Word\_array = []

   Count = []

   while( i < len(Str)) :

      word = ""

      while(Str[i] != ' ') :

         word += Str[i]

         i = i + 1

         if( i == len(Str)) :

             break

      if(i != len(Str)) :

         while(Str[i] == ' ') :

            i = i + 1

      if(len(Word\_array) == 0) :

         Word\_array.append(word)

         Count.append(1)

      else :

         flag = 1

         for j in range(len(Word\_array)) :

            if(Word\_array[j] == word) :

               Count[j] += 1

               flag = 0

               break

         if (flag == 1) :

            Word\_array.append(word)

            Count.append(1)

   for i in range(len(Word\_array)) :

      print("\t%15s : %d "%(Word\_array[i],Count[i]))

def main():

   while True :

      print ("\t\t  \*\*\*\* STRING OPERATIONS \*\*\*\*")

      print ("\t\t1 : Display word with longest length")

      print ("\t\t2 : Determine the frequency of occurrence of particular character in the string")

      print ("\t\t3 : Check whether given string is palindrome or not ")

      print ("\t\t4 : Display index of first appearance of the substring")

      print ("\t\t5 : Count the occurrences of each word in a given string")

      print ("\t\t6 : Exit")

      ch = int(input("Enter your choice : "))

      if (ch == 6):

         print ("End of Program")

         quit()

      elif (ch == 1) :

         Display\_word\_with\_longest\_length()

      elif (ch == 2) :

         Determine\_frequency\_of\_occurrence\_of\_particular\_character\_in\_string()

      elif (ch == 3) :

         Check\_for\_palindrome()

      elif (ch == 4) :

         display\_index\_of\_first\_appearance\_of\_the\_substring()

      elif (ch == 5) :

         Count\_\_occurrences\_of\_each\_word\_in\_given\_string()

      else :

         print ("Wrong choice entered !! Try again")

main()

--------------------OUTPUT----------------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a3.py

        \*\*\*\* STRING OPERATIONS \*\*\*\*

      1 : Display word with longest length

      2 : Determine the frequency of occurrence of particular character in the string

      3 : Check whether given string is palindrome or not

      4 : Display index of first appearance of the substring

      5 : Count the occurrences of each word in a given string

      6 : Exit

Enter your choice : 1

Enter the main string : man is walking

   Word with longest length is walking having lenght 7

        \*\*\*\* STRING OPERATIONS \*\*\*\*

      1 : Display word with longest length

      2 : Determine the frequency of occurrence of particular character in the string

      3 : Check whether given string is palindrome or not

      4 : Display index of first appearance of the substring

      5 : Count the occurrences of each word in a given string

      6 : Exit

Enter your choice : 2

Enter the string : what is there

Enter the character  : 2

   String : what is there

   Character : 2

   Frequency of occurrence of character(2) in string(what is there) is 0

        \*\*\*\* STRING OPERATIONS \*\*\*\*

      1 : Display word with longest length

      2 : Determine the frequency of occurrence of particular character in the string

      3 : Check whether given string is palindrome or not

      4 : Display index of first appearance of the substring

      5 : Count the occurrences of each word in a given string

      6 : Exit

Enter your choice : 3

Enter the string to be checked : what is there

   what is there string is not an palindrome string

        \*\*\*\* STRING OPERATIONS \*\*\*\*

      1 : Display word with longest length

      2 : Determine the frequency of occurrence of particular character in the string

      3 : Check whether given string is palindrome or not

      4 : Display index of first appearance of the substring

      5 : Count the occurrences of each word in a given string

      6 : Exit

Enter your choice : 4

Enter the main string : man is walking

Enter the sub string to check : what is there

Main String : man is walking

Substring String : what is there

Substring not found in the main string

        \*\*\*\* STRING OPERATIONS \*\*\*\*

      1 : Display word with longest length

      2 : Determine the frequency of occurrence of particular character in the string

      3 : Check whether given string is palindrome or not

      4 : Display index of first appearance of the substring

      5 : Count the occurrences of each word in a given string

      6 : Exit

Enter your choice : 5

Enter the main string : man is walking

               man : 1

                is : 1

           walking : 1

        \*\*\*\* STRING OPERATIONS \*\*\*\*

      1 : Display word with longest length

      2 : Determine the frequency of occurrence of particular character in the string

      3 : Check whether given string is palindrome or not

      4 : Display index of first appearance of the substring

      5 : Count the occurrences of each word in a given string

      6 : Exit

Enter your choice : 6

End of Program

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ '''

**Practical 4.** ‘’’Write a python program that computes the net amount of a bank account

based a transaction log from console input. The transaction log format

is shown as following: D 100 W 200 (Withdrawal is not allowed if balance

is going negative. Write functions for withdraw and deposit) D means deposit

while W means withdrawal.

Suppose the following input is supplied to the program:

D 300

D 300

W 200

D 100

Then, the output should be: 500’’’

#function to perform deposite operation by adding amount to the balance

def deposit(balance,amount):

    return balance+amount

#function to perform withdrawl operation by subtracting amount from the balace

#only alow withdrawl if the balance is sufficient

def withdrawl(balance,amount):

     if balance>=amount:

       return balance-amount

     else:

       print(f"Insufficient balance to withdraw {amount} current balance:{balance}")

       return balance

#function to manually convert a string to an integer without using inbuilt function

def string\_to\_int(s):

    number=0

    for char in s:

        number=number \*10+(ord(char)-ord('0'))

        return number

#function to manually handle input reading from the console without using inbuilt input() function

def get\_input(prompt):

    print(prompt,end="")

    input\_string=""

    while True:

        char=input()

        if char=="":

               break

        input\_string +=char

        return input\_string

#function to process the transaction log and calculate the final balance

def process\_transaction(transaction\_log):

    balance=0

    i=0

    lenght=0

#Manually calculate the lenght of the transaction log string

    lenght += 1

#Iterate over each character in the transaction log

    while i<lenght:

       action=transaction\_log[i]

       #Get the action (D or W)

       amount\_str=""

       while i<lenght and transaction\_log[i]!=',':

          amount\_str+=transaction\_log[i]

          i+=1

#manually trim reading and trailing spaces from the amount string\

       start\_index=0

       end\_index=0

       while amount\_str[start\_index]=='':

             start\_index +=1

       end\_index=start\_index

    while end\_index<len(amount\_str) and amount\_str[end\_index]!='':

       end\_index +=1

    amount=string\_to\_int(amount\_str[start\_index:end\_index])

#convert amount string to integer

#perform deposit or withdrawl based on the action

    if action == 'D':

             balance=deposit(balance,amount)

    elif action == 'W':

             balance=withdraw(balance,amount)

    i+=2

    return balance

#return the final balance after processing all transaction

#input account details from user

account\_holder\_name=input("Enter account holder's name:")

account\_number=input("Enter account num:")

ifsc\_code=input("Enter ifsc code:")

mobile\_number=input("Enter mobile number:")

#inout transaction log from the user

transaction\_log=input("Enter transaction log:")

#compute the net amount by processing the transaction log

net\_balance=process\_transaction(transaction\_log)

#output the account details and the final balance

print("\n Account Details:")

print(f"Account Holder's name:{account\_holder\_name}")

print(f"Account Numer:{account\_number}")

print(f"IFSC Code:{ifsc\_code}")

print(f"Mobile Number:{mobile\_number}")

print(f"Net Balance:{net\_balance}")

---------------output-------------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a4.py

Enter account holder's name:a

Enter account num:123

Enter ifsc code:234

Enter mobile number:234567890

Enter transaction log:D100

 Account Details:

Account Holder's name:a

Account Numer:123

IFSC Code:234

Mobile Number:234567890

Net Balance:20

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ '''

**# Practicle No 5:** write a python program to store roll no 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 sentinal search.

def accept\_array(A):

    n=int(input("Enter the total no of student:"))

    for i in range(n):

        x=int(input("Enter the roll no of student %d:"%(i+1)))

        A.append(x)

    print("student info accepted successfully\n\n")

    return n

def display\_array(A,n):

    if(n==0):

        print("\n No records in the database")

    else:

        print("student Array:",end=' ')

        for i in range(n):

            print("%d "%A[i],end=' ')

        print("\n");

def Linear\_Search(A,n,x):

    for i in range(n):

        if(A[i]==x):

            return i

    return -1

def Sentinel\_search(A,n,x):

    last=A[n-1]

    i=0

    A[n-1]=x

    while(A[i]!=x):

        i=i+1

    A[n-1]=last

    if((i<n-1) or (x==A[n-1])):

        return i

    else:

        return -1

def Main():

    A=[]

    while True:

        print("\t1:Accept&Display student info")

        print("\t2:Linear search")

        print("\t3:sentinel search")

        print("\t4:Exit")

        ch=int(input("Enter your choice:"))

        if(ch==4):

            print("End of program")

            quit()

        elif(ch==1):

            A=[]

            n=accept\_array(A)

            display\_array(A,n)

        elif(ch==2):

            x=int(input("Enter the roll\_no to be searched:"))

            flag= Linear\_Search(A,n,x)

            if(flag== -1):

                print("\t Roll no to be searched not found\n")

            else:

                print("\t Roll no found at location %d"%(flag+1))

        elif(ch==3):

            X=int(input("Enter the roll\_no to be searched:"))

            flag=Sentinel\_search(A,n,x)

            if(flag== -1):

                print("\t Rol no to be searched not found\n")

            else:

                print("\t Roll no found at location%d"%(flag+1))

        else:

            print("Wrong choice entered !! Try again")

Main()

 #-----------OUTPUT--------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a5.py

    1:Accept&Display student info

    2:Linear search

    3:sentinel search

    4:Exit

Enter your choice:1

Enter the total no of student:2

Enter the roll no of student 1:1

Enter the roll no of student 2:2

student info accepted successfully

student Array: 1  2

    1:Accept&Display student info

    2:Linear search

    3:sentinel search

    4:Exit

Enter your choice:2

Enter the roll\_no to be searched:1

     Roll no found at location 1

    1:Accept&Display student info

    2:Linear search

    3:sentinel search

    4:Exit

Enter your choice:3

Enter the roll\_no to be searched:2

     Roll no found at location1

    1:Accept&Display student info

    2:Linear search

    3:sentinel search

    4:Exit

Enter your choice:4

End of program

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$

# **Practicle No 6:** write a python program to store roll no of student in 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.

def accept\_array(A):

   n = int(input("Enter the total no. of student : "))

   print("Input roll numbers in sorted order")

   for i in range(n):

      x = int(input("Enter the  roll no of student %d : "%(i+1)))

      A.append(x)

   print("Student Info accepted successfully\n\n")

   return n

def display\_array(A,n):

   if(n == 0) :

      print("\nNo records in the database")

   else :

      print("Students  Array : ",end=' ')

      for i in range(n) :

         print("%d  "%A[i],end=' ')

      print("\n");

def Recursive\_Binary\_Search(A,s,l,X) :

   if(s <= l ) :

      mid = int((s + l) / 2)

      if(A[mid] == X) :

         return mid      # Found

      else :

         if(X < A[mid] ) :

            return Recursive\_Binary\_Search(A,s,mid-1,X)

         else :

            return Recursive\_Binary\_Search(A,mid+1,l,X)

   return -1 # NOT FOUND

def Iterative\_Binary\_Search(A,n,X) :

   s = 0

   l = n-1

   while(s <= l ) :

      mid = int((s + l) / 2)

      if(A[mid] == X) :

         return mid       # Found

      else :

         if (X < A[mid] )  :

            l = mid-1

         else :

            s = mid+1

   return -1; #NOT FOUND

#Returns index of x if present,  else returns -1

def Fibonacci\_Search(A,n,X) :

   f1 = 0

   f2 = 1

   f3 = f1 + f2

   offset = -1

   while (f3 < n) :

      f1 = f2

      f2 = f3

      f3  = f1 + f2

   while (f3 > 1) :

      i = min(offset+f1, n-1)

      if(A[i] == X) :

         return i        #Found

      else :

         if (X < A[i] ) :   # left substudent (66 % or 2/3 student)

            f3  = f1

            f2 = f2 - f1

            f1 = f3 - f2

         else :     # right substudent ( 33 % or 1/3 student)

            f3  = f2

            f2 = f1

            f1 = f3 - f2

            offset = i

   if(f2 == 1 and (offset+1) < n and A[offset + 1] == X) :

      return offset+1      # Found

   return -1    #NOT FOUND

def Main() :

   A = []

   while True :

      print ("\t1 : Accept & Display Students info ")

      print ("\t2 : Recursive Binary Search")

      print ("\t3 : Iterative Binary Search")

      print ("\t4 : Fibonacci Search")

      print ("\t5 : Exit")

      ch = int(input("Enter your choice : "))

      if (ch == 5):

         print ("End of Program")

         quit()

      elif (ch==1):

         A = []

         n = accept\_array(A)

         display\_array(A,n)

      elif (ch==2):

         X = int(input("Enter the roll\_no to be searched : "))

         flag  = Recursive\_Binary\_Search(A,0,n-1,X)

         if(flag == -1) :

            print("\tRoll no to be Searched not Found\n")

         else :

            print("\tRoll no found at location %d"%(flag + 1))

      elif (ch==3):

         X = int(input("Enter the roll\_no to be searched : "))

         flag  = Iterative\_Binary\_Search(A,n,X)

         if(flag == -1) :

            print("\tRoll no to be Searched not Found\n")

         else :

            print("\tRoll no found at location %d"%(flag + 1))

      elif (ch==4):

         X = int(input("Enter the roll\_no to be searched : "))

         flag  = Fibonacci\_Search(A,n,X)

         if(flag == -1) :

            print("\tRoll no to be Searched not Found\n")

         else :

            print("\tRoll no found at location %d"%(flag + 1))

      else :

           print ("Wrong choice entered !! Try again")

Main()

---------OUTPUT-----------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a6.py

   1 : Accept & Display Students info

   2 : Recursive Binary Search

   3 : Iterative Binary Search

   4 : Fibonacci Search

   5 : Exit

Enter your choice : 1

Enter the total no. of student : 3

Input roll numbers in sorted order

Enter the  roll no of student 1 : 1

Enter the  roll no of student 2 : 23

Enter the  roll no of student 3 : 4

Student Info accepted successfully

Students  Array :  1   23   4

   1 : Accept & Display Students info

   2 : Recursive Binary Search

   3 : Iterative Binary Search

   4 : Fibonacci Search

   5 : Exit

Enter your choice : 3

Enter the roll\_no to be searched : 2

   Roll no to be Searched not Found

   1 : Accept & Display Students info

   2 : Recursive Binary Search

   3 : Iterative Binary Search

   4 : Fibonacci Search

   5 : Exit

Enter your choice : 4

Enter the roll\_no to be searched : 23

   Roll no found at location 2

   1 : Accept & Display Students info

   2 : Recursive Binary Search

   3 : Iterative Binary Search

   4 : Fibonacci Search

   5 : Exit

Enter your choice : 5

End of Program

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ '''

# **Practicle No 7:** write a python program to store first year percentage of student in array write a function for sorting array of floating point number in asending order using  selection sort and bubble sort and display top five scores

def accept\_array(A):

   n = int(input("Enter the total no. of student : "))

   for i in range(n):

      x = float(input("Enter the  first year percentage of student %d : "%(i+1)))

      A.append(x)

   print("Array accepted successfully\n\n");

def display\_array(A):

   n = len(A)

   if(n == 0) :

      print("\nNo records in the database")

   else :

      print("Array of FE Marks : ",end=' ')

      for i in range(n) :

         print("%.2f  "%A[i],end=' ')

      print("\n");

def Selection\_sort(A) :

   n = len(A)

   for pos in range(n-1):

      min\_ind = pos

      for i in range(pos + 1, n) :

         if(A[i] < A[min\_ind]) :

            min\_ind = i

      temp = A[pos]

      A[pos] = A[min\_ind]

      A[min\_ind] = temp

def Bubble\_sort(A) :

   n = len(A)

   for Pass in range(1,n) :

      for i in range(n-Pass) :

         if(A[i] < A[i+1]) :

            temp = A[i]

            A[i] = A[i+1]

            A[i+1] = temp

def Main() :

   A = []

   while True :

      print ("\t1 : Accept & Display the FE Marks")

      print ("\t2 : Selection Sort Ascending order")

      print ("\t3 : Bubble sort Descending order and display top five scores")

      print ("\t4 : Exit")

      ch = int(input("Enter your choice : "))

      if (ch == 4):

         print ("End of Program")

         quit()

      elif (ch==1):

         accept\_array(A)

         display\_array(A)

      elif (ch==2):

         print("Marks before sorting")

         display\_array(A)

         Selection\_sort(A)

         print("Marks after sorting")

         display\_array(A)

      elif (ch==3):

         print("Marks before sorting")

         display\_array(A)

         Bubble\_sort(A)

         print("Marks after sorting")

         display\_array(A)

         if(len(A) >= 5) :

            print("Top Five Scores : ")

            for i in range(5) :

               print("\t%.2f"%A[i])

         else :

            print("Top Scorers : ")

            for i in range(len(A)) :

               print("\t%.2f"%A[i])

      else :

           print ("Wrong choice entered !! Try again")

Main()

------------OUTPUT--------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a7.py

   1 : Accept & Display the FE Marks

   2 : Selection Sort Ascending order

   3 : Bubble sort Descending order and display top five scores

   4 : Exit

Enter your choice : 1

Enter the total no. of student : 2

Enter the  first year percentage of student 1 : 78

Enter the  first year percentage of student 2 : 90

Array accepted successfully

Array of FE Marks :  78.00   90.00

   1 : Accept & Display the FE Marks

   2 : Selection Sort Ascending order

   3 : Bubble sort Descending order and display top five scores

   4 : Exit

Enter your choice : 2

Marks before sorting

Array of FE Marks :  78.00   90.00

Marks after sorting

Array of FE Marks :  78.00   90.00

   1 : Accept & Display the FE Marks

   2 : Selection Sort Ascending order

   3 : Bubble sort Descending order and display top five scores

   4 : Exit

Enter your choice : 3

Marks before sorting

Array of FE Marks :  78.00   90.00

Marks after sorting

Array of FE Marks :  90.00   78.00

Top Scorers :

   90.00

   78.00

   1 : Accept & Display the FE Marks

   2 : Selection Sort Ascending order

   3 : Bubble sort Descending order and display top five scores

   4 : Exit

Enter your choice : 4

End of Program

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$

**# Practicle No 8:** write a python program to store second year percentage of student in array of floating point number in ascending order using a)insertion sort b)shell sort and display top five score

def accept\_array(A):

   n = int(input("Enter the total no. of student : "))

   for i in range(n):

      x = float(input("Enter the  Second year percentage of student %d : "%(i+1)))

      A.append(x)

   print("Array accepted successfully\n\n");

def display\_array(A):

   n = len(A)

   if(n == 0) :

      print("\nNo records in the database")

   else :

      print("Array of SE Marks : ",end=' ')

      for i in range(n) :

         print("%.2f  "%A[i],end=' ')

      print("\n");

def Insertion\_sort(A) :

   n = len(A)

   for i in range(1,n) :

      element = A[i]

      j  = i-1

      while( j >= 0) :

         if (A[j] <= element) :

            break

         else :

            A[j+1] = A[j]

            j = j - 1

      A[j+1] = element

def Insertion\_sort\_gap(A,n,gap,s) :

   for i in range(s+gap,n,gap) :

      element = A[i]

      j = i - gap

      while( j >=0 ) :

         if(A[j] <= element) :

            break;

         else :

            A[j+gap] = A[j]

            j = j - gap

      A[j+gap] = element;

def Shell\_sort(A) :

   n = len(A)

   gap = int(n / 2)

   while( gap > 0) :

      for s in range(gap) :

         Insertion\_sort\_gap(A,n,gap,s)

      gap = int(gap / 2)

def Main() :

   A = []

   while True :

      print ("\t1 : Accept & Display the SE Marks")

      print ("\t2 : Insertion Sort Ascending order")

      print ("\t3 : Shell sort Ascending order and display top five scores")

      print ("\t4 : Exit")

      ch = int(input("Enter your choice : "))

      if (ch == 4):

         print ("End of Program")

         quit()

      elif (ch==1):

         A = []

         accept\_array(A)

         display\_array(A)

      elif (ch==2):

         print("Marks before sorting")

         display\_array(A)

         Insertion\_sort(A)

         print("Marks after sorting")

         display\_array(A)

      elif (ch==3):

         print("Marks before sorting")

         display\_array(A)

         Shell\_sort(A)

         print("Marks after sorting")

         display\_array(A)

         n =len(A)

         if(n >= 5) :

            print("Top Five Scores : ")

            for i in range(n-1,n-6,-1) :

               print("\t%.2f"%A[i])

         else :

            print("Top Scorers : ")

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

               print("\t%.2f"%A[i])

      else :

           print ("Wrong choice entered !! Try again")

Main()

------------OUTPUT--------------

'''gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ python3 a8.py

   1 : Accept & Display the SE Marks

   2 : Insertion Sort Ascending order

   3 : Shell sort Ascending order and display top five scores

   4 : Exit

Enter your choice : 1

Enter the total no. of student : 3

Enter the  Second year percentage of student 1 : 56

Enter the  Second year percentage of student 2 : 89

Enter the  Second year percentage of student 3 : 67

Array accepted successfully

Array of SE Marks :  56.00   89.00   67.00

   1 : Accept & Display the SE Marks

   2 : Insertion Sort Ascending order

   3 : Shell sort Ascending order and display top five scores

   4 : Exit

Enter your choice : 2

Marks before sorting

Array of SE Marks :  56.00   89.00   67.00

Marks after sorting

Array of SE Marks :  56.00   67.00   89.00

   1 : Accept & Display the SE Marks

   2 : Insertion Sort Ascending order

   3 : Shell sort Ascending order and display top five scores

   4 : Exit

Enter your choice : 3

Marks before sorting

Array of SE Marks :  56.00   67.00   89.00

Marks after sorting

Array of SE Marks :  56.00   67.00   89.00

Top Scorers :

   89.00

   67.00

   56.00

   1 : Accept & Display the SE Marks

   2 : Insertion Sort Ascending order

   3 : Shell sort Ascending order and display top five scores

   4 : Exit

Enter your choice : 4

End of Program

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$

/\* Practicle No 9: 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>

using namespace std;

class list;

class node

{

    int prn;

    string name;

    node \*next;

public:

    node(int x,string nm)

{

        prn=x;

        next=NULL;

        name=nm;

}

friend class list;

};

class list

{

    node \*start;

public:

    list(){

        start=NULL;

    }

    void create();

    void display();

    void insertAtBeginning();

    void insertAtEnd();

    void insertAfter();

    void deleteAtFirst();

    void deleteByValue();

    void deleteAtEnd();

    int computeTotal();

    void sortList();

    void concatList(list &q1);

    void displayRev(node \*t);

    bool reverseDisplay()

    {

        if(start==NULL)

            return false;

        node \*temp=start;

        displayRev(temp);

            return true;

            }

};

void list::displayRev(node \*t)

{

    if(t==NULL)

        return;

    else

    {

        displayRev(t->next);

        cout<<"\n PRN No:"<<t->prn<<"Name:"

        <<t->name;

    }

}

void list::create()

{

    int no;

    string nam;

    if(start==NULL)

    {

        cout<<"Enter PRN number:";

        cin>>no;

        cout<<"Enter name:";

        cin>>nam;

        cout<<nam;

        start=new node(no,nam);

        cout<<"\n====List created====";

    }

    else

    {

        cout<<"\n List is already created";

    }

}

void list::display()

{

    node \*t;

    t=start;

    if(start==NULL)

        cout<<"\nList is empty";

    else

    {

        cout<<"\n ====list====\n";

            while (t!=NULL){

                cout<<t->prn<<"  "<<t->name<<"\n";

                    t=t->next;

            }

    }

}

void list::insertAtBeginning()

{

    int no;

    string name;

    node \*temp;

    if(start==NULL)

    {

        create();

    }

    else

    {

        cout<<"\n Enter PRN nuber:";

        cin>>no;

        cout<<"Enter name:";

        cin>>name;

        //cout<<num;

        temp=new node(no,name);

        temp->next=start;

        start=temp;

        cout<<"Inserted"<<temp->name<<"at the beginning:";

    }

}

void list::insertAtEnd()

{

    int no;

    string nam;

    node \*t;

    if(start==NULL)

        create();

    else

    {

        cout<<"\n Enter PRN number:";

        cin>>no;

        cout<<"Enter name:";

        cin>>nam;

        t=start;

        while(t->next !=NULL)

            t=t->next;

            node \*p=new node(no,nam);

            t->next=p;

    }

}

void list::insertAfter()

{

    int prev\_no;

    cout<<"\n Enter PRN No of after do you want insert:";

    cin>>prev\_no;

    node \*t;

    t=start;

    string nam;

    int flag=0,no;

    while(t!=NULL)

    {

        if(t->prn==prev\_no)

        {

            flag=1;break;

        }

        t=t->next;

    }

    if(flag==1)

    {

        node \*p;

        cout<<"\n Enter PRN number:";

        cin>>no;

        cout<<"Enter name:";

        cin>>nam;

        p=new node(no,nam);

        p->next=t->next;

        t->next=p;

    }

    else

    {

        cout<<"\n"<<prev\_no<<"is not in list:";

    }

}

void list::deleteAtFirst()

{

    node \*t;

    if(start==NULL)

        cout<<"\nClub is Empty..";

    else

    {

    t=start;

    start=start->next;

    t->next=NULL; //Not necessary

    delete t;

    cout<<"\nPresident deleted..";

}

}

void list::deleteByValue()

{

    int no,flag=0;

    node \*t,\*prev;

    if(start==NULL)

        cout<<"\nList/Club is empty;";

    else

    {

        cout<<"\nEnter PRN no. of member to be deleted: ";

        cin>>no;

        t=start->next; //t=start if we have to delete precident also.. start->next is first member

        while(t->next!=NULL)

        {

            if(t->prn==no){

                flag=1;

                break;

            }

            prev=t;

            t=t->next;

        }

        if(flag==1)

        {

            prev->next=t->next;

            t->next=NULL;

            delete t;

            cout<<"\nMember with prn no: "<<no<<" is deleted.";

        }

        else

            cout<<"\nMember not found in List./president or secretary cannot be deleted.";

    }

}

void list::deleteAtEnd()

{

    node \*t,\*prev;

    t=start;

        if(start==NULL)

        cout<<"\nClub is Empty..";

    else

    {

    while(t->next!=NULL)

    {

        prev=t;

        t=t->next;

    }

    prev->next=NULL;

    delete t;

    cout<<"\nSecretary Deleted.";

}

}

int list::computeTotal()

{

node \*t;

int count=0;

t=start;

if(start==NULL)

{

    cout<<"\nList is mpty.";

    return 0;

}

while(t!=NULL)

{

count++;

t=t->next;

}

return count;

}

void list::sortList()

{

    node \*i,\*j,\*last=NULL;

    int tprn;

    string tname;

if(start==NULL)

{

    cout<<"\nList is empty.";

    return ;

}

for(i=start;i->next!=NULL;i=i->next)

{

    for(j=start;j->next!=last;j=j->next)

    {

        if((j->prn)>(j->next->prn))

        {

            tprn=j->prn;

            tname=j->name;

            j->prn=j->next->prn;

            j->name=j->next->name;

            j->next->prn=tprn;

            j->next->name=tname;

        }

    }

}

cout<<"\n List is sorted.";

display();

}

void list::concatList(list &q1)

{

    node \*t,\*p;

    t=q1.start;

    if(t==NULL)

    {

        cout<<"\nList 2 is empty";

        return;

    }

    p=start; //first list

    while(p->next!=NULL)

    {

        p=p->next;

    }

    p->next=t;

    q1.start=NULL; //second list is set to  null

    cout<<"\nAfter concatenationlist";

    display();

}

int main() {

    list \*l;

    int choice,selectList;

    list l1,l2;

    l=&l1;

    X:cout<<"\nSelect List\n1.List 1\n2.List 2\nEnter choice: ";

    cin>>selectList;

    if(selectList==1)

    {

        l=&l1;

    }

    else if(selectList==2)

    {

        l=&l2;

    }

    else

    {

        cout<<"\nWrong list Number.";

        goto X;

    }

    do

    {

        cout<<"\n1. create\n2.Insert President\n3.Insert secretary\n4.insert after position(member)\n5.Display list"

        <<"\n6.Delete President\n7.Delete Secretary\n8.Delete Member\n9.Find total No. of members\n10.Sort list\n11. Reselect List ++--##"

        <<"\n12.Combine lists\n13.Reverse Display\n0. Exit\nENter your choice:\t";

        cin>>choice;

        switch(choice)

        {

            case 1: l->create();

                break;

            case 2: l->insertAtBeginning();

                break;

            case 3: l->insertAtEnd();

                break;

            case 4: l->insertAfter();

                break;

            case 5: l->display();

                break;

            case 6: l->deleteAtFirst();

                break;

            case 7: l->deleteAtEnd();

                break;

            case 8: l->deleteByValue();

                break;

            case 9: cout<<"\nTotal members(including President & Secretary): "<<l->computeTotal();

                break;

            case 10: l->sortList();

            break;

            case 11:

                goto X;

                break;

                case 12:

                    l1.concatList(l2);

                    break;

                case 13:

                    l->reverseDisplay();

                    break;

            deafult:

                 cout<<"Wrong choice";

        }

    }while(choice!=0);

cout<<"\n========== GOOD BYE ====================\n";

    return 0;

}

----------------OUTPUT----------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a9.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

Select List

1.List 1

2.List 2

Enter choice: 1

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  1

Enter PRN number:2345

Enter name:a

a

====List created====

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  2

 Enter PRN nuber:123

Enter name:w

Insertedwat the beginning:

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  3

 Enter PRN number:123

Enter name:h

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  4

 Enter PRN No of after do you want insert:123

 Enter PRN number:456

Enter name:q

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  5

 ====list====

123  w

456  q

2345  a

123  h

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  6

President deleted..

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  7

Secretary Deleted.

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  8

Enter PRN no. of member to be deleted: 456

Member not found in List./president or secretary cannot be deleted.

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  9

Total members(including President & Secretary): 2

1. create

2.Insert President

3.Insert secretary

4.insert after position(member)

5.Display list

6.Delete President

7.Delete Secretary

8.Delete Member

9.Find total No. of members

10.Sort list

11. Reselect List ++--##

12.Combine lists

13.Reverse Display

0. Exit

ENter your choice:  10

 List is sorted.

 ====list====

456  q

2345  a\*/

/\* Practicle No 10: 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

1) Set of students who like both Vanilla and Butterscotch

2) Set of students who like either Vanilla or Butterscotch or not both

3) 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 roll no: ";

     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 roll no  ";

          cout<<"\n  2. To enter the roll no of student who like vanila";

          cout<<"\n  3.  To enter the roll no of student who like butterscotch";

          cout<<"\n  4.  To display the roll no of student who like vanila or butterscotch";

          cout<<"\n  5.  To display the roll no of student who like only vanila";

          cout<<"\n  6.  To display the roll no of student who like only butterscotch";

          cout<<"\n  7.  To display the roll no of student who like both vanila and butterscotch ";

          cout<<"\n  8.  To display the roll no of student who neither like vanila nor butterscotch";

          cout<<"\nEnter a choice: ";

          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;

}

--------------OUTPUT--------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a10.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 1

enter no. of students: 2

enter student roll no: 1

enter student roll no: 2

1

2

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 2

enter no. of  students who like vanila: 1

enter student roll no: 2

2

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 3

enter no. of students who like butterscotch: 1

enter student roll no: 1

1

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 3

enter no. of students who like butterscotch: 2

enter student roll no: 1

enter student roll no: 2

1

2

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 4

students who like vanila or butterscotch:

2

1

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 5

students  like only vanila :

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 6

students like only butterscotch:

1

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 7

students who like both vanila and butterscotch:

2

 do you want to continue enter y/Y :

y

 choice the options

  1. To enter all students roll no

  2. To enter the roll no of student who like vanila

  3.  To enter the roll no of student who like butterscotch

  4.  To display the roll no of student who like vanila or butterscotch

  5.  To display the roll no of student who like only vanila

  6.  To display the roll no of student who like only butterscotch

  7.  To display the roll no of student who like both vanila and butterscotch

  8.  To display the roll no of student who neither like vanila nor butterscotch

Enter a choice: 8

students who like neither vanila nor butterscotch:

 do you want to continue enter y/Y : \*/

/\* **Practicle No 11:** write cpp program for storing binary number using doubly linked list write function 1)To compute 1's and 2's complement 2) Add two binary number\*/

#include<iostream>

using namespace std;

class binary;

class node

{

    node \*prev;

    bool n;

    node \*next;

public:

    node()

    {

        prev=next=NULL;

    }

    node(bool b)

    {

        n=b;

        prev=next=NULL;

    }

    friend class binary;

};

class binary

{

    node \*start;

    public:

        binary()

        {

            start=NULL;

        }

        void generateBinary(int no);

        void displayBinary();

        void onesComplement();

        void twoscomplement();

        binary operator +(binary n1);

            bool addBitAtBegin(bool val)

           {

        node \*nodee=new node(val);

         if(start==NULL)

        {

            start=nodee;

        }

        else

        {

            nodee->next=start;

            start->prev=nodee;

            start=nodee;

        }

        return true;

    }

};

void binary::generateBinary(int no)

{

    bool rem;

    node \*p;

    rem=no%2;

    start=new node(rem);

    no=no/2;

    while(no!=0)

    {

        rem=no%2;

        no=no/2;

    /\*

        if(start==NULL)

        {

            start=new node(rem);

        //  cout<<" Start prev: "<<start->prev;

        //  cout<<" Start next: "<<start->next ;

        }

        else

        {

        \*/

            p=new node(rem);

            p->next=start;

            start->prev=p;

        //  cout<<" Start prev: "<<start->prev->n;

        //  cout<<"   p->n"<<p->n;

            start=p;

        //}

    }

}

void binary::displayBinary()

{

    node \*t;

    t=start;

    while(t!=NULL)

    {

        cout<<t->n;

        t=t->next;

    }

}

void binary::onesComplement()

{

    node \*t;

    t=start;

    while(t!=NULL)

    {

        if(t->n==0)

            t->n=1;

        else

            t->n=0;

        t=t->next;

    }

}

binary binary::operator +(binary n1)

{

    binary sum;

    node \*a=start;

    node \*b=n1.start;

//  bit \*s=sum.start;

    bool carry=false;

    while(a->next!=NULL)

        a=a->next;

    while(b->next!=NULL)

        b=b->next;

    while(a!=NULL && b!=NULL)

    {

        sum.addBitAtBegin((a->n)^(b->n)^carry);

        carry=((a->n&& b->n) || (a->n&& carry) || (b->n && carry));

        a=a->prev;

        b=b->prev;

    }

    while(a!=NULL)

    {

        sum.addBitAtBegin(a->n^carry);

        a=a->prev;

    }

    while(b!=NULL)

    {

        sum.addBitAtBegin(b->n^carry);

        b=b->prev;

    }

    sum.addBitAtBegin(carry);

    return sum;

}

void binary::twoscomplement()

{

    onesComplement();

    bool carry=1;

    node \*t;

    t=start;

    while(t->next!=NULL)

    {

        t=t->next;

    }

    while(t!=NULL)

    {

    if(t->n==1&& carry==1)

    {

        t->n=0;

        carry=1;

    }

    else

    if(t->n==0&& carry==1)

    {

        t->n=1;

        carry=0;

    }

    else

    if(carry==0)

    break;

    t=t->prev;

}

displayBinary();

}

int main()

{

    int num,num1;

    binary n1,n3,n2;

    int choice=1;

    do

    {

        cout<<"\n\n=========Binary Number Operations========\n";

        cout<<"1. Generate binary\n2.One's Complement\n3.Two's Complement\n4. Addition\n0.Exit\nEnter your choice: ";

        cin>>choice;

        switch(choice)

        {

            case 1: cout<<"\nENter Number in decimal form: ";

                    cin>>num;

                    n1.generateBinary(num);

                    cout<<"\nBinary Representation: ";

                    n1.displayBinary();

                    break;

            case 2:cout<<"\nENter Number in decimal form: ";

                    cin>>num;

                    n1.generateBinary(num);

                    cout<<"\nBinary Representation: ";

                    n1.displayBinary();

                    cout<<"\nOnes Complement: ";

                    n1.onesComplement();

                    n1.displayBinary();

                    break;

            case 3:cout<<"\nENter Number in decimal form: ";

                    cin>>num;

                    n1.generateBinary(num);

                    cout<<"\nBinary Representation: ";

                    n1.displayBinary();

                    cout<<"\nTwos complement; ";

                    n1.twoscomplement();

                    break;

            case 4: cout<<"\nENter Two Numbers: ";

                    cin>>num>>num1;

                    n1.generateBinary(num);

                    n2.generateBinary(num1);

                    n1.displayBinary();

                    cout<<" + ";

                    n2.displayBinary();

                    cout<<"= ";

                    n3=n1+n2;

                    n3.displayBinary();

                                      }

    }while(choice!=0);

    n1.generateBinary(7);

    cout<<"\nBinary Representation: ";

    n1.displayBinary();

//

//  cout<<"\nOnes Complement: ";

//  n1.displayBinary();

    cout<<"\nTwos complement; ";

    n1.twoscomplement();

    return 0;

}

----------OUTPUT-----------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a11.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

=========Binary Number Operations========

1. Generate binary

2.One's Complement

3.Two's Complement

4. Addition

0.Exit

Enter your choice: 1

ENter Number in decimal form: 12

Binary Representation: 1100

=========Binary Number Operations========

1. Generate binary

2.One's Complement

3.Two's Complement

4. Addition

0.Exit

Enter your choice: 2

ENter Number in decimal form: 1

Binary Representation: 1

Ones Complement: 0

=========Binary Number Operations========

1. Generate binary

2.One's Complement

3.Two's Complement

4. Addition

0.Exit

Enter your choice: 3

ENter Number in decimal form: 3

Binary Representation: 11

Twos complement; 01

=========Binary Number Operations========

1. Generate binary

2.One's Complement

3.Two's Complement

4. Addition

0.Exit

Enter your choice: 4

ENter Two Numbers: 12 1

1100 + 1= 01101

=========Binary Number Operations========

1. Generate binary

2.One's Complement

3.Two's Complement

4. Addition

0.Exit

Enter your choice: 0

Binary Representation: 111

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ \*/

/\* Practicle No 12: 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-

1. To check whether given string is palindrome or not that uses a stack to determine whether

a string is a palindrome.

2. to remove spaces and punctuation in string, convert all the Characters to lowercase, and

then call above Palindrome checking function to check for a palindrome

3. to print string in reverse order using stack.

\*/

#include<iostream>

#include<string.h>

#define MAX 100

using namespace std;

 struct stack

 {

     char data[MAX];

     int top;

 };

 void init(stack \*stck)

 {

     int i;

     for(i=0;i<MAX;i++)

     {

         stck->data[i]='\0';

         stck->top=-1;

     }

 }

 void print(stack stck)

 {

     int i;

     cout<<"\n Stack elements are";

     for(i=0;i<MAX;i++)

     {

         cout<<"stck.data[i]";

         cout<<"\t top="<<stck.top;

     }

 }

 int isempty(stack stck)

 {

     return stck.top==-1?1:0;

 }

 int isfull(stack stck)

 {

     return stck.top==MAX-1?1:0;

 }

 void push(stack \*stck,char data[MAX])

 {

     int i;

     for(i=0;data[i]!='\0';i++)

     {

     stck->top+=1;

     stck->data[stck->top]=data[i];

     }

 }

 void push1(stack \*stck,char data[MAX])

 {

     int i;

     for(i=0;data[i]!='\0';i++)

     {

       if(data[i]!=' ')

       {

     stck->top+=1;

     stck->data[stck->top]=data[i];

       }

     }

 }

 void pop(stack \*stck)

 {

     char rev[MAX];

     int i=0;

     while(!isempty(\*stck))

     {

     rev[i]=stck->data[stck->top];

     stck->top-=1;

     i++;

     }

     rev[i]='\0';

     cout<<"\n Original string is="<<stck->data;

     cout<<"\n reverse string is=";

     cout<<rev;

     if(strcmp(rev,stck->data))

     {

     cout<<"\n string is not palindrome";

     }

     else

     {

     cout<<"\n String is palindrome";

     }

 }

 int main()

 {

     stack stck;

     char data[MAX]={'E','v','a',',',' ','c','a','n',' ','I',' ','s','e','e',' ','b','e','e','s',' ','i','n',' ','a',' ','c','a','v','e','\0'};

     init(&stck);

     int ch,i,j=0;

     while(ch!=3)

     {

     cout<<"\n1.Push the string;Print reverse string & check palindrome or not(pop)";

     cout<<"\n2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not";

     cout<<"\n3.Exit";

     cout<<"\n Enter your choice";

     cin>>ch;

     switch(ch)

     {

     case 1:

         if(isfull(stck))

         {

         cout<<"\n Stack is full";

         }

         else

         {

         push(&stck,data);

         }

         if(isempty(stck))

         {

         cout<<"\n Stack is empty";

         }

         else

         {

         pop(&stck);

         }

         break;

     case 2:

         init(&stck);

         if(isfull(stck))

         {

         cout<<"\n Stack is full";

         }

         else

         {

         for(i=0;data[i]!='\0';i++)

          {

            if(data[i]!=' ')

            {

             if(data[i]>=65 && data[i]<=90)

              {

                data[i]=data[i]+32;

              }

            }

          }

        push1(&stck,data);

         }

         pop(&stck);

         break;

         }

     }

     return (0);

 }

 ------------OUTPUT------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a12.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

1.Push the string;Print reverse string & check palindrome or not(pop)

2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not

3.Exit

 Enter your choice1

 Original string is=Eva, can I see bees in a cave

 reverse string is=evac a ni seeb ees I nac ,avE

 string is not palindrome

1.Push the string;Print reverse string & check palindrome or not(pop)

2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not

3.Exit

 Enter your choice2

 Original string is=eva,caniseebeesinacave

 reverse string is=evacaniseebeesinac,ave

 string is not palindrome

1.Push the string;Print reverse string & check palindrome or not(pop)

2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not

3.Exit

 Enter your choice3

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ \*/

/\* Practicle No 12: 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-

1. To check whether given string is palindrome or not that uses a stack to determine whether

a string is a palindrome.

2. to remove spaces and punctuation in string, convert all the Characters to lowercase, and

then call above Palindrome checking function to check for a palindrome

3. to print string in reverse order using stack.

\*/

#include<iostream>

#include<string.h>

#define MAX 100

using namespace std;

 struct stack

 {

     char data[MAX];

     int top;

 };

 void init(stack \*stck)

 {

     int i;

     for(i=0;i<MAX;i++)

     {

         stck->data[i]='\0';

         stck->top=-1;

     }

 }

 void print(stack stck)

 {

     int i;

     cout<<"\n Stack elements are";

     for(i=0;i<MAX;i++)

     {

         cout<<"stck.data[i]";

         cout<<"\t top="<<stck.top;

     }

 }

 int isempty(stack stck)

 {

     return stck.top==-1?1:0;

 }

 int isfull(stack stck)

 {

     return stck.top==MAX-1?1:0;

 }

 void push(stack \*stck,char data[MAX])

 {

     int i;

     for(i=0;data[i]!='\0';i++)

     {

     stck->top+=1;

     stck->data[stck->top]=data[i];

     }

 }

 void push1(stack \*stck,char data[MAX])

 {

     int i;

     for(i=0;data[i]!='\0';i++)

     {

       if(data[i]!=' ')

       {

     stck->top+=1;

     stck->data[stck->top]=data[i];

       }

     }

 }

 void pop(stack \*stck)

 {

     char rev[MAX];

     int i=0;

     while(!isempty(\*stck))

     {

     rev[i]=stck->data[stck->top];

     stck->top-=1;

     i++;

     }

     rev[i]='\0';

     cout<<"\n Original string is="<<stck->data;

     cout<<"\n reverse string is=";

     cout<<rev;

     if(strcmp(rev,stck->data))

     {

     cout<<"\n string is not palindrome";

     }

     else

     {

     cout<<"\n String is palindrome";

     }

 }

 int main()

 {

     stack stck;

     char data[MAX]={'E','v','a',',',' ','c','a','n',' ','I',' ','s','e','e',' ','b','e','e','s',' ','i','n',' ','a',' ','c','a','v','e','\0'};

     init(&stck);

     int ch,i,j=0;

     while(ch!=3)

     {

     cout<<"\n1.Push the string;Print reverse string & check palindrome or not(pop)";

     cout<<"\n2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not";

     cout<<"\n3.Exit";

     cout<<"\n Enter your choice";

     cin>>ch;

     switch(ch)

     {

     case 1:

         if(isfull(stck))

         {

         cout<<"\n Stack is full";

         }

         else

         {

         push(&stck,data);

         }

         if(isempty(stck))

         {

         cout<<"\n Stack is empty";

         }

         else

         {

         pop(&stck);

         }

         break;

     case 2:

         init(&stck);

         if(isfull(stck))

         {

         cout<<"\n Stack is full";

         }

         else

         {

         for(i=0;data[i]!='\0';i++)

          {

            if(data[i]!=' ')

            {

             if(data[i]>=65 && data[i]<=90)

              {

                data[i]=data[i]+32;

              }

            }

          }

        push1(&stck,data);

         }

         pop(&stck);

         break;

         }

     }

     return (0);

 }

 ------------OUTPUT-----------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a12.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

1.Push the string;Print reverse string & check palindrome or not(pop)

2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not

3.Exit

 Enter your choice1

 Original string is=Eva, can I see bees in a cave

 reverse string is=evac a ni seeb ees I nac ,avE

 string is not palindrome

1.Push the string;Print reverse string & check palindrome or not(pop)

2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not

3.Exit

 Enter your choice2

 Original string is=eva,caniseebeesinacave

 reverse string is=evacaniseebeesinac,ave

 string is not palindrome

1.Push the string;Print reverse string & check palindrome or not(pop)

2.Remove blank space,convert upperspace into lowercase,push the string & check palindrome or not

3.Exit

 Enter your choice3

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ \*/

/\* **Practicle No 13:** Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions:

Operands and operator, both must be single character.

Input Postfix expression must be in a desired format.

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

#include<iostream>

#include<stack>

using namespace std;

bool isOperator(char c)

{

if(c=='+'||c=='-'||c=='\*'||c=='/'||c=='^')

{

return true;

}

else

{

return false;

}

}

int precedence(char c)

{

 if(c == '^')

 return 3;

 else if(c == '\*' || c == '/')

 return 2;

 else if(c == '+' || c == '-')

 return 1;

 else

 return -1;

}

string InfixToPostfix(stack<char> s, string infix)

{

string postfix;

for(int i=0;i<infix.length();i++)

{

if((infix[i] >= 'a' && infix[i] <= 'z')

||(infix[i] >= 'A' && infix[i] <= 'Z'))

{

postfix+=infix[i];

}

else if(infix[i] == '(')

{

s.push(infix[i]);

}

else if(infix[i] == ')')

{

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

{

char temp=s.top();

postfix+=temp;

s.pop();

}

if(s.top()=='(')

{

s.pop();

}

}

else if(isOperator(infix[i]))

{

if(s.empty())

{

s.push(infix[i]);

}

else

{

if(precedence(infix[i])>precedence(s.top()))

{

s.push(infix[i]);

}

else if((precedence(infix[i])==precedence(s.top()))&&(infix[i]=='^'))

{

s.push(infix[i]);

}

else

{

while((!s.empty())&&( precedence(infix[i])<=precedence(s.top())))

{

postfix+=s.top();

s.pop();

}

s.push(infix[i]);

}

}

}

}

while(!s.empty())

{

postfix+=s.top();

s.pop();

}

return postfix;

}

int main()

{

 string infix\_exp, postfix\_exp;

 cout<<"Enter a Infix Expression :"<<endl;

 cin>>infix\_exp;

 stack <char> stack;

cout<<"INFIX EXPRESSION: "<<infix\_exp<<endl;

 postfix\_exp = InfixToPostfix(stack, infix\_exp);

 cout<<endl<<"POSTFIX EXPRESSION: "<<postfix\_exp;

return 0;

}

------------OUTPUT---------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a13.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

Enter a Infix Expression :

a+b

INFIX EXPRESSION: a+b

POSTFIX EXPRESSION: ab+gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ \*/

/\* **Practicle No 14:** 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 5

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:";

cin>>x;

obj.enqueue(x);

  }

         else

     cout<< "Queue is overflow";

          break;

  case 2: if(!obj.isempty())

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

   else

{   cout<<"\n Queue is underflow";  }

   cout<<"\nremaining jobs :";

   obj.display();

          break;

 case 3: if (!obj.isempty())

       {  cout<<"\n Queue contains:";

      obj.display();

       }

       else

        cout<<"\n Queue is empty";

      break;

 case 4: cout<<"\n Exit";

        }

      }while(ch!=4);

return 0;

}

-----------OUTPUT-------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a14.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

 1. insert job

 2.delete job

 3.display

 4.Exit

 Enter your choice:1

 Enter data:12

 1. insert job

 2.delete job

 3.display

 4.Exit

 Enter your choice:2

 Deleted Element=12

remaining jobs :

 1. insert job

 2.delete job

 3.display

 4.Exit

 Enter your choice:3

 Queue is empty

 1. insert job

 2.delete job

 3.display

 4.Exit

 Enter your choice:4

 Exitgescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ \*/

/\*Practicle No 15: Write program to implement a priority queue in C++ using an inorder list to store the items in the queue. Create a class that includes the data items (which should be template) and the priority (which should be int). The inorder list should contain these objects, with operator <= overloaded so that the items with highest priority appear at the start of the list (which will make it relatively easy to retrieve the highest item.)\*/

#include <iostream>

#include <cstdio>

#include <cstring>

#include <cstdlib>

using namespace std;

typedef struct n {    // node declaration

   int p;

   int info;

   struct n \*l;

}n1;

class Priority\_Queue

{

   private:

               //Declare a front pointer f and initialize it to NULL.

      n1 \*f;

   public:

   Priority\_Queue() //constructor

     {

         f = NULL;

      }

   void insert(int i, int p)

     {

         n1 \*t, \*q;

         t = new n;

         t->info = i;

         t->p = p;

         if (f == NULL || p < f->p)

   {

            t->l= f;

            f = t;

         }

   else

   {

            q = f;

            while (q->l != NULL && q->l->p <= p)

               q = q->l;

               t->l = q->l;

               q->l = t;

         }

      }

    void del()

    {

         n1 \*t;

         if(f == NULL) //if queue is null

            cout<<"Queue Underflow\n";

         else

          {

            t = f;

            cout<<"Deleted item is: "<<t->info<<endl;

            f = f->l;

            free(t);

         }

      }

      void show() //print queue

      {

    n1 \*ptr;

         ptr = f;

         if (f == NULL)

            cout<<"Queue is empty\n";

         else

          {

            cout<<"Queue is :\n";

            cout<<"Priority Item\n";

            while(ptr != NULL) {

               cout<<ptr->p<<" "<<ptr->info<<endl;

               ptr = ptr->l;

            }

         }

      }

};

int main() {

   int c, i, p;

   Priority\_Queue pq;

   do //perform switch opeartion

   {

      cout<<"1.Insert\n";

      cout<<"2.Delete\n";

      cout<<"3.Display\n";

      cout<<"4.Exit\n";

      cout<<"Enter your choice : ";

      cin>>c;

      switch(c) {

         case 1:

            cout<<"Input the item value to be added in the queue : ";

            cin>>i;

            cout<<"Enter its priority : ";

            cin>>p;

            pq.insert(i, p);

            break;

         case 2:

            pq.del();

            break;

         case 3:

            pq.show();

            break;

         case 4:

            break;

         default:

         cout<<"Wrong choice\n";

      }

   }

   while(c != 4);

   return 0;

}

-------------OUTPUT------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a15.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice : 1

Input the item value to be added in the queue : 12

Enter its priority : 2

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice : 2

Deleted item is: 12

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice : 3

Queue is empty

1.Insert

2.Delete

3.Display

4.Exit

Enter your choice : 4

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ \*/

**/\*Practicle No 16:** 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<stdlib.h>

#include<unistd.h>

using namespace std;

int front=0,rear=-1;

int cnt=0;

int data[5];

void enque(int no)

{

rear=(rear+1)%5;

data[rear]=no;

cnt++;

}

int isfull()

{

if(cnt==5)

return 1;

else

return 0;

}

void display()

{

int i;

for(i=front;i!=rear;i=(i+1)%5)

{

cout<<"\nOrder of table no  has been served"<<data[i];

sleep(2);

}

cout<<"\nOrder of table no has been served"<<data[i];

sleep(2);

}

int main()

{

int ch,n;

cout<<"\nMaximum orders served are 5";

do

{

cout<<"\n\nMenu\n1.Place an order\n2.Display the serving list";

cin>>ch;

switch(ch)

{

case 1: if(!isfull())

    {

        cout<<"\nEnter your table no.";

        cin>>n;

        enque(n);

        }

    else

    cout<<"\nOrders Full!!!!";

    break;

case 2: display();

    break;

}

}while(ch!=3);

return 0;

}

--------------OUTPUT-------------

/\*gescoe@gescoe-OptiPlex-7010:~$ cd Desktop

gescoe@gescoe-OptiPlex-7010:~/Desktop$ cd SEA-9

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ g++ a16.cpp

gescoe@gescoe-OptiPlex-7010:~/Desktop/SEA-9$ ./a.out

Maximum orders served are 5

Menu

1.Place an order

2.Display the serving list1

Enter your table no.2

Menu

1.Place an order

2.Display the serving list1

Enter your table no.4

Menu

1.Place an order

2.Display the serving list2

Order of table no  has been served2

Order of table no has been served4\*/