**Assignment 6:**

**Input:**

#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 add\_data(pb):

roll=int(input("Enter roll number"))

name=input("Enter name..")

flag=False

for i in range(len(pb)):

if roll < pb[i][0]:

pb.insert(i, (roll, name))

flag = True

break

if not flag:

pb.append((roll,name))

return pb

def display(pb):

for c in pb:

print(c)

def Ternary\_Search(pb):

roll\_no=int(input("Enter roll no search:"))

length=len(pb)

l=0

r=length-1

flag=0

while(l<=r):

# Calculate mid1 and mid2

mid1 = l + (r - l) // 3

mid2 = r - (r - l) // 3

# Check if the element is present at the mid1 or mid2 positions

if roll\_no==pb[mid1][0]:

print("roll number=",roll\_no,"and name is:",pb[mid1][1])

print(pb[mid1][1], "is member of club....")

flag=1

break

elif roll\_no==pb[mid2][0]:

print("roll number=",roll\_no,"and name is:",pb[mid2][1])

print(pb[mid2][1], "is member of club....")

flag=1

break

# If x is in the left one-third

elif roll\_no< pb[mid1][0]:

r=mid1-1

flag=0

# If x is in the right one-third

elif roll\_no > pb[mid2][0]:

l=mid2+1

flag=0

# If x is in the middle one-third

else:

l=mid1+1

r-=mid2-1

flag=0

if flag==0:

print("Roll number",roll\_no,"is not in club....add the name")

add\_data(pb)

display(pb)

if \_\_name\_\_=="\_\_main\_\_":

pb=[]

print("1.Insert roll number and name....")

print("2.display roll number and name....")

print("3.Search name...")

while(True):

print("enter choice:")

ch=int(input())

if ch==1:

pb=add\_data(pb)

elif ch==2:

display(pb)

elif ch==3:

Ternary\_Search(pb)

else:

print("wrong choice...")

break

**Output:**

1.Insert roll number and name....

2.display roll number and name....

3.Search name...

enter choice:

1

Enter roll number2

Enter name..ram

enter choice:

2

(2, 'ram')

enter choice:

3

Enter roll no search:2

roll number= 2 and name is: ram

ram is member of club.

**Assignment 7 :**

**Input:**

#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 add\_Elements(arr,n):

flag=False

print("Enter Array Elements :")

for i in range(0,n):

num=int(input())

arr.append(num)

print("Array Elements are:")

print(arr)

return arr

def bubble\_sort(arr,n):

for i in range(0,n):

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

if(arr[j]>arr[j+1]):

temp=arr[j]

arr[j]=arr[j+1]

arr[j+1]=temp

print("sorted elements using bubble sort....")

print(arr)

def selection\_sort(arr,n):

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

for j in range(i+1,n):

if(arr[i]>arr[j]):

temp=arr[i]

arr[i]=arr[j]

arr[j]=temp

print("sorted elements using selection sort....")

print(arr)

if \_\_name\_\_=="\_\_main\_\_":

arr=[]

n=0

print("1.Insert array elements....")

print("2.display array elements....")

print("3.Bubble sort...")

print("3.Selection sort...")

while(True):

print("enter choice:")

ch=int(input())

if ch==1:

n=int(input("how many number do you want to add..."))

arr=add\_Elements(arr,n)

elif ch==2:

bubble\_sort(arr,n)

elif ch==3:

selection\_sort(arr,n)

else:

print("wrong choice...")

break

**Output:**

1.Insert array elements....

2.display array elements....

3.Bubble sort...

3.Selection sort...

enter choice:

1

how many number do you want to add...5

Enter Array Elements :

20

40

10

30

50

Array Elements are:

[20, 40, 10, 30, 50]

enter choice:

2

sorted elements using bubble sort....

[10, 20, 30, 40, 50]

enter choice:

3

sorted elements using selection sort....

[10, 20, 30, 40, 50]

**Assignment 8:**

**Input:**

//Write C++ program to maintain club

//member 壮 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<vector>

Using namespace std;

Class node{

Public:

Int prn;

String name;

Node \*next;

Node(){

Next = nullptr;

}

};

Class List{

Node \*head;

Public:

List()

{

Head = NULL;

}

Void insertAtBeg(int v,string s)

{

Node \*p = new node();

p->prn = v;

p->name=s;

p->next = head;

head = p;

}

Void insertAtEnd(int v,string s)

{

Node \*p = new node();

p->prn = v;

p->name=s;

if (head == NULL)

{

Head = p;

}

Else

{

Node \*ptr = head;

While (ptr->next != NULL)

{

Ptr = ptr->next;

}

Ptr->next = p;

}

}

Void display()

{

Int count=0;

If (head == NULL){

Cout<<”List is empty”<<endl;

}

Else

{

Node \*p = head;

Cout<<”Linked List: \n”;

While (p != NULL){

Cout<<p->prn<<”\t”<<p->name<<”\n”;

Count++;

P = p->next;

}

}

Cout<<”Number of elemnets in linked list:”<<count;

}

Void insertBetween(int v,string s, int pos){

Node \*p = new node();

p->prn = v;

p->name=s;

if (pos == 0)

{

p->next = head;

head = p;

}

Else

{

Node \*ptr = head;

While(pos>1)

{

Ptr = ptr->next;

--pos;

}

p->next = ptr->next;

ptr->next = p;

}

}

Void deleteAtFirst()

{

If (head == NULL){

Cout<<”List is Empty”<<endl;

}

Else

{

Cout<<”Element Deleted: “<<head->prn<<”\t”<<head->name<<endl;

Node \*p = head;

Head = head->next;

Delete(p);

}

}

Void deleteAtEnd()

{

If (head == NULL){

Cout<<”List is Empty”<<endl;

}

Else if (head->next == NULL)

{

Cout<<”Element Deleted: “<<head->prn<<”\t”<<head->name<<endl;

Delete(head);

Head = NULL;

}

Else

{

Node \*p = head;

While (p->next->next != NULL)

{

P = p->next;

}

Cout<<”Element Deleted: “<<p->next->prn<<endl;

Delete(p->next);

p->next = NULL;

}

}

Void deleteInBetween(int pos)

{

If (head == NULL)

{

Cout<<”List is Empty”<<endl;

}

Else

{

Node \*p, \*ptr;

If (pos == 0)

{

Cout<<”Element Deleted: “<<head->prn<<”\t”<<head-

>name<<endl;

Ptr = head;

Head = head->next;

Delete(ptr);

}

Else

{

P = ptr = head;

While(pos>0)

{

--pos;

P = ptr;

Ptr = ptr->next;

}

Cout<<”Element Deleted: “<<ptr->prn<<”\t”<<ptr-

>name<<endl;

p->next = ptr->next;

free(ptr);

}

}

}

};

Int main()

{

Printf(“1 to Insert at the beginning”);

Printf(“\n2 to Insert at the end”);

Printf(“\n3 to Insert at mid”);

Printf(“\n4 to Delete from beginning”);

Printf(“\n5 to Delete from the end”);

Printf(“\n6 to Delete from mid”);

Printf(“\n7 to Display”);

Printf(“\n0 to Exit”);

Int choice,v,p;

String nm;

List ll;

Do

{

Cout<<”\nEnter Your Choice: “;

Cin>>choice;

Switch (choice)

{

Case 1:

Cout<<”Enter Element: “;

Cin>>v;

Cout<<”Enter name: “;

Cin>>nm;

ll.insertAtBeg(v,nm);

break;

case 2:

cout<<”Enter Element: “;

cin>>v;

cout<<”Enter name: “;

cin>>nm;

ll.insertAtEnd(v,nm);

break;

case 3:

cout<<”Enter Element: “;

cin>>v;

cout<<”Enter name: “;

cin>>nm;

cout<<”Enter Position ( zero-indexed ): “;

cin>>p;

ll.insertBetween(v,nm,p);

break;

case 4:

ll.deleteAtFirst();

break;

case 5:

ll.deleteAtEnd();

break;

case 6:

cout<<”Enter Position ( zero-indexed ): “;

cin>>p;

ll.deleteInBetween(p);

break;

case 7:

ll.display();

break;

}

} while (choice != 0);

}

**Output:**

1 to Insert at the beginning

2 to Insert at the end

3 to Insert at mid

4 to Delete from beginning

5 to Delete from the end

6 to Delete from mid

7 to Display

0 to Exit

Enter Your Choice: 1

Enter Element: 10

Enter name: Soham

Enter Your Choice: 2

Enter Element: 20

Enter name: Srujan

Enter Your Choice: 3

Enter Element: 13

Enter name: Sanket

Enter Position ( zero-indexed ): 1

Enter Your Choice: 4

Element Deleted: 10 Soham

Enter Your Choice: 3

Enter Element: 30

Enter name: Geetesh

Enter Position ( zero-indexed ): 1

Enter Your Choice: 6

Enter Position ( zero-indexed ): 1

Element Deleted: 30 Geetesh

Enter Your Choice: 5

Element Deleted: 20

Enter Your Choice: 7

Linked List:

13 Sanket

Number of elemnets in linked list:1

**Assignment 9:**

**Input:**

//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>

#include<vector>

using namespace std;

class node{

public:

string name;

char van,but;

node \*next;

node(){

next = nullptr;

}

};

class List

{

node \*head;

public:

List()

{

head = NULL;

}

void insertAtEnd(string s,char v,char b)

{

node \*p = new node();

p->van = v;

p->name=s;

p->but=b;

if (head == NULL)

{

head = p;

}

else

{

node \*ptr = head;

while (ptr->next != NULL)

{

ptr = ptr->next;

}

ptr->next = p;

}

}

void display()

{

int count=0;

if (head == NULL){

cout<<"List is empty"<<endl;

}

else

{

node \*p = head;

cout<<"Linked List: \n";

while (p != NULL){

cout<<p->name<<"\t"<<p->van<<"\t"<<p->but<<"\n";

count++;

p = p->next;

}

}

cout<<"Number of elemnets in linked list:"<<count;

}

void disp\_vanAndbut()

{

//int count=0;

if (head == NULL){

cout<<"List is empty"<<endl;

}

else

{

node \*p = head;

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

while (p != NULL)

{

if(p->van=='y' && p->but=='y')

{

cout<<p->name<<"\t"<<p->van<<"\t"<<p->but<<"\n";

//count++;

}

p = p->next;

}

}

//cout<<"Number of elemnets in linked list:"<<count;

}

void disp\_EitherVanOrBut()

{

//int count=0;

if (head == NULL){

cout<<"List is empty"<<endl;

}

else

{

node \*p = head;

cout<<"Students who like Either vanila Or Butterscotch.......\n";

while (p != NULL)

{

if(!((p->van=='y' && p->but=='y')||(p->van=='n' && p->but=='n')))

{

cout<<p->name<<"\n";

//count++;

}

p = p->next;

}

}

//cout<<"Number of elemnets in linked list:"<<count;

}

void disp\_NitherVanNorBut()

{

//int count=0;

if (head == NULL){

cout<<"List is empty"<<endl;

}

else

{

node \*p = head;

cout<<"Students who like Either vanila Or Butterscotch.......\n";

while (p != NULL)

{

if(p->van=='n' && p->but=='n')

{

cout<<p->name<<"\n";

//count++;

}

p = p->next;

}

}

//cout<<"Number of elemnets in linked list:"<<count;

}

};

int main()

{

printf("1 to Insert data...");

printf("\n2 Display data.....");

printf("\n3 Display Students who like both vanila and butterscotch......");

printf("\n4 Display Students who like Either Vanila Or Butterscotch......");

printf("\n4 Display Students who like Nither Vanila Nor Butterscotch......");

printf("\n0 to Exit");

int choice,p;

char van,but;

string nm;

List ll;

do

{

cout<<"\nEnter Your Choice: ";

cin>>choice;

switch (choice)

{

case 1:

cout<<"Enter name: ";

cin>>nm;

cout<<"Do you like vanilla y/n: ";

cin>>van;

cout<<"Do you like butterscotch y/n: ";

cin>>but;

ll.insertAtEnd(nm,van,but);

break;

case 2:

ll.display();

break;

case 3:

ll.disp\_vanAndbut();

break;

case 4:

ll.disp\_EitherVanOrBut();

break;

case 5:

ll.disp\_NitherVanNorBut();

break;

}

} while (choice != 0);

}

**Assignment 10:**

**Input:**

Write c++ program with functions:

To print original string followed by reversed string using stack and

To check given string is Palindrome or not…..

\*/

#include<iostream>

Using namespace std;

#include<string>

#define max 50

Char stk[max];

Class stack

{

Int top=-1;

Public:

Void push(string ch)

{

If(top==max-1)

{

Cout<<”Stack is full…..\n”;

}

Else

{

For (int i=0;i<ch.length();i++)

{

Top++;

Stk[top]=ch[top];

}

}

}

String pop(string str)

{

String rev=””;

Cout<<stk;

If(top==-1)

Cout<<”Stack is Empty….”;

Else

{

For(int i=top;i>=0;i--)

{

Cout<<stk[top];

Rev=rev+stk[top];

Top--;

}

}

Cout<<”\n Top is:”<<top;

Return rev;

}

};

Int main()

{

String str,rev,str1;

Int c;

Stack s1;

Do

{

Cout<<”\n\n Select from Menu………..”;

Cout<<”\n1. Print Original String followed by reversed String using Stack…..”;

Cout<<”\n2: To check given string is Palindrome or not…..”;

Cout<<”\n3: Press 0 to Exit……”;

Cout<<”\nEnter your choice:”;

Cin>>c;

Switch©

{

Case 1:

Cout<<”\nEnter String:”;

Cin>>str;

Cout<<”\nString is: “;

S1.push(str);

Rev=s1.pop(str);

Break;

Case 2:

Cout<<”\nOriginal String:”<<str;

Cout<<”\nReversed String :”<<rev;

If(str==rev)

{

Cout<<”\nString “<<str<<” is Palindrome……”;

}

Else

Cout<<”\nString “<<str<<” is not Palindrome……”;

Break;

}

}while(c!=0);

}

**Output:**

Select from Menu...........

1. Print Original String followed by reversed String using Stack.....

2: To check given string is Palindrome or not.....

3: Press 0 to Exit......

Enter your choice:1

Enter String:nitin

String is: nitinnitin

Top is:-1

Select from Menu...........

1. Print Original String followed by reversed String using Stack.....

2: To check given string is Palindrome or not.....

3: Press 0 to Exit......

Enter your choice:2

Original String:nitin

Reversed String :nitin

String nitin is Palindrome......

Select from Menu...........

1. Print Original String followed by reversed String using Stack.....

2: To check given string is Palindrome or not.....

3: Press 0 to Exit......

Enter your choice:0

[Program finished