**ASSIGNMENT 1**

**AIM:**

To create ADT that implement the "set" concept.

a. Add (new Element) -Place a value into the set

b. Remove (element)

c. Contains (element) Return true if element is in collection

d. Size () Return number of values in collection

e. Intersection of two sets

f. Union of two sets

g. Difference between two sets

h. Subset

**CODE**

#include<iostream>

#include<stdlib.h>

using namespace std;

void create(int set[])

{

int n;

cout<<"\n enter the size of set : ";

cin>>n;

cout<<"\n enter the elements in the set : ";

for(inti=1;i<=n;i++)

cin>>set[i];

set[0]=n;

}

bool member(int set[],intnum)

{

for(inti=1;i<=set[0];i++)

if(set[i]==num)

return true;

return false;

}

void intersection(int set1[],int set2[],int set3[])

{

for(inti=1;i<=set2[0];i++)

{

if(member(set1,set2[i])== true)

{

set3[0]++;

set3[set3[0]]=set2[i];

}

}

}

void union1(int set1[],int set2[],int set4[])

{

for(inti=0;i<=set1[0];i++)

set4[i]=set1[i];

for(inti=1;i<=set2[0];i++)

{

if(member(set1,set2[i])== false)

{

set4[0]++;

set4[set4[0]]=set2[i];

}

}

}

void difference1(int set1[],int set2[],int set5[])

{

for(inti=1;i<=set1[0];i++)

{

if((member(set2,set1[i]) == false))

{

set5[0]++;

set5[set5[0]]=set1[i];

}

; }

}

void contains(int set[])

{

intnum;

cout<<"\n enter the element to be searched ";

cin>>num;

if((member(set,num))== true)

cout<<"\n element is present ";

else

cout<<"\n element is not present ";

}

bool subset(int seta[],intsetb[])

{

for(inti=1;i<=setb[0];i++)

{

if((member(seta,setb[i]))==true)

continue;

else

return false;

}

return true;

}

void remove(int set[])

{

intpos;

cout<<"\n enter the position from which you want to remove the element : ";

cin>>pos;

if(pos<=set[0])

{

if(pos<set[0])

{

for(inti=pos;i<=set[0];i++)

{

set[i]=set[i+1];

}

set[0]--;

}

else if(pos==set[0])

{

set[0]--;

}

}

else

{

cout<<"\n entered position exceeds the size of the set " ;

}

}

void size(int set[])

{

cout<<set[0];

}

void display(int set[])

{

cout<<"\n size : "<<set[0]<<"\t";

for(inti=1;i<=set[0];i++)

{

cout<<set[i]<<" ";

}

}

int main()

{

int set1[10];

cout<<"\n FOR SET 1 ";

create(set1);

int set2[10];

cout<<"\n FOR SET 2 ";

create(set2);

intch,c;

char choice;

do{

cout<<"\n\n -------------- OPERATION MENU ---------------- ";

cout<<"\n 1 for INTERSECTION ";

cout<<"\n 2 for UNION ";

cout<<"\n 3 for DIFFERENCE ";

cout<<"\n 4 for CONTAINS( if element is present in set or not)";

cout<<"\n 5 for SUBSET";

cout<<"\n 6 for REMOVE";

cout<<"\n 7 for SIZE";

cout<<"\n 8 for DISPLAY";

cout<<"\n 9 for EXIT";

cout<<"\n\n Enter your choice : ";

cin>>ch;

switch(ch)

{

case 1:

{

int set3[1];

set3[0]=0;

cout<<"\n the intersection of two sets : \t";

intersection(set1,set2,set3);

display(set3);

break;

}

case 2:

{

int set4[set1[0]+1];

set4[0]=0;

cout<<"\n the union of two sets \t";

union1(set1,set2,set4);

display(set4);

break;

}

case 3:

{

int set5[1];

set5[0]=0;

cout<<"\n the difference of two sets \t";

difference1(set1,set2,set5);

display(set5);

break;

}

case 4:

{

cout<<"\n enter 1 for searching in set1 and 2 for searching in set2 ";

cin>>c;

switch(c)

{

case 1: contains(set1); break;

case 2: contains(set2); break;

default: cout<<"\n wrong choice entered ";

}

break;

}

case 5:

{

label:

cout<<"\n enter 1 for checking if set1 is subset of set2 else enter 2 ";

cin>>c;

switch(c)

{

case 1:

{

if(subset(set2,set1)==true )

cout<<"\n set1 is subset of set2";

else

cout<<"\n set1 is not a subset of set2";

break;

}

case 2:

{

if(subset(set1,set2)==true )

cout<<"\n set2 is subset of set1";

else

cout<<"\n set2 is not a subset of set1";

break;

}

default:

cout<<"\n wrong choice entered ";

goto label;

}

break;

}

case 6:

{

label2:

cout<<"\n enter 1 for removing element from set 1 and 2 for removal from set 2 ";

cin>>c;

switch(c)

{

case 1:

{

remove(set1);

break;

}

case 2:

{

remove(set2);

break;

}

default:

cout<<"\n wrong choice entered ";

goto label2;

}

break;

}

case 7:

{

cout<<"\n size of set 1 : ";

size(set1);

cout<<"\n size of set 2 : ";

size(set2);

break;

}

case 8:

{

display(set1);

display(set2);

break;

}

case 9:

exit(0);

default:

cout<<"\n wrong choice entered ";

}

cout<<"\n want to continue with the operation ?(y/n) :";

cin>>choice;

}while((choice=='y')||(choice=='Y'));

return 0;

}

OUTPUT:

