**Roll No-82**

**Assignment No:-**

**Assignment Name:-Write a program to implement Union and Find Operation(Modify).**

#include"iostream.h"

#include"conio.h"

class SET

{

int n,\*PAR;

public:

SET(int);

void UNION(int,int);

int FIND(int);

void DISPLAY();

};

SET::SET(int par)

{

n=par;

PAR=new int[n+1];

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

PAR[i]=-1;

}

void SET::UNION(int i,int j)

{

if(i!=j)

{

if(FIND(i) != FIND(j) )

{

int x=PAR[i]+PAR[j];

if(PAR[i]<PAR[j])

{

PAR[j]=i;

PAR[i]=x;

}

else

{

PAR[i]=j;

PAR[j]=x;

}

return;

}

}

cout<<"\nUnion not possible";

}

int SET::FIND(int i)

{

int j=i;

while(PAR[j]>0)

j=PAR[j];

return j;

}

void SET::DISPLAY()

{

cout<<"\nNODE\tPARENT\tWT";

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

cout<<endl<<i<<"\t"<<PAR[i];

}

void MENU()

{

int opt,size,i,j;

cout<<"\nEnter no of sets : ";

cin>>size;

SET obj(size);

do

{

cout<<"\n1 Union";

cout<<"\n2 Find";

cout<<"\n3 DIsp";

cout<<"\n4 Exit";P

cout<<"\nEnter your option : ";

cin>>opt;

switch(opt)

{

case 1:

cout<<"\nEnter roots of two sets : ";

cin>>i>>j;

obj.UNION(i,j);

break;

case 2:

cout<<"\nEnter ele to find : ";

cin>>i;

cout<<"\nRoot of "<<i<<" is "<<obj.FIND(i);

break;

case 3:

obj.DISPLAY();

break;

case 4:

return;

default:

cout<<"\ninvalid input";

break;

}

}

while(1);

}

void main()

{

clrscr();

MENU();

getch();

}