DISJOINT SET

#include <stdio.h>

struct DisjSet {

int parent[10];

int rank[10];

int n;

}dis; void makeSet()

{

for (int i = 0; i < dis.n; i++) {

dis.parent[i] = i;

dis.rank[i]=0;

}

}

void displaySet()

{ printf("\nParent Array\n");

for (int i = 0; i < dis.n; i++) {

printf("%d ",dis.parent[i]); }

printf("\nRank Array\n");

for (int i = 0; i < dis.n; i++)

{

printf("%d ",dis.rank[i]); }

printf("\n");

} int find(int x)

{

if (dis.parent[x] != x) {

dis.parent[x] = find(dis.parent[x]);

}

return dis.parent[x];

}

void Union(int x, int y)

{

int xset = find(x);

int yset = find(y);

if (xset == yset)

return;

if (dis.rank[xset] < dis.rank[yset]) { dis.parent[xset] = yset;

dis.rank[xset]=-1;

}

else if (dis.rank[xset] > dis.rank[yset]) { dis.parent[yset] = xset;

dis.rank[yset]=-1;

} else {

dis.parent[yset] = xset; dis.rank[xset] = dis.rank[xset] + 1; dis.rank[yset]=-1;

}

}

int main() { int n,x,y;

printf("How many elements ?"); scanf("%d",&dis.n); makeSet(); int ch,wish; do

{

printf("\n\_\_\_MENU\_\n");

printf("1. Union \n2.Find\n3.Display\n");

printf("enter choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("Enter elements to perform union");

scanf("%d %d",&x,&y);

Union(x, y);

break;

case 2: printf("Enter elements to check if connected components");

scanf("%d %d",&x,&y);

if (find(x) == find(y))

printf("Connected components\n") ;

else

printf("Not onnected components \n") ;

break;

case 3: displaySet();

break;

}

printf("\nDo you wish to continue ?(1/0)\n");

scanf("%d",&wish);

}while(wish==1);

return 0;

}