**Swiss Pairing**

Description -> A Swiss-system tournament is a non-eliminating tournament format which features a set number of rounds of competition. The first tournament of this type was a chess tournament in Zurich in 1895, hence the name "Swiss system". Swiss systems are commonly used in chess, bridge, Scrabble, backgammon, and many other games.

Objective ->To manage chess tournament by Matching Players according to their Rating and Scores.

Source Code:

#include<stdio.h>

#include<conio.h>

#include<string.h>

struct players

{

char nm[20];

int rate;

int score;

int round;

}p[50];

void display(struct players p[],int n)

{

int i=0;

for(i=0;i<n;i++)

printf("\n%d.%-8s\t%-4d",i+1,p[i].nm,p[i].rate);

}

void sort(struct players p[],int n)

{

int i,j;

char temp[20]=""; //for swaping name

int srate,sscore; //for swaping rate and score

if(p[0].round==0)

{

for(i=0;i<n-1;i++)

{

for(j=0;j<n-1;j++)

{

if((p[j].rate)<(p[j+1].rate))

{

strcpy(temp,p[j+1].nm);

strcpy(p[j+1].nm,p[j].nm);

strcpy(p[j].nm,temp);

srate=p[j+1].rate;

p[j+1].rate=p[j].rate;

p[j].rate=srate;

}

}

}

}

if(p[0].round>0)

{

for(i=0;i<n-1;i++)

{

for(j=0;j<n-1;j++)

{

if((p[j].score)<(p[j+1].score))

{

strcpy(temp,p[j+1].nm);

strcpy(p[j+1].nm,p[j].nm);

strcpy(p[j].nm,temp);

srate=p[j+1].rate;

p[j+1].rate=p[j].rate;

p[j].rate=srate;

sscore=p[j+1].score;

p[j+1].score=p[j].score;

p[j].score=sscore;

}

}

}

}

}

void result(struct players p[],int n)

{

int r=0,i; //for adding result score

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

printf("\t\nEnter Sr. No of players who won");

for(i=1;i<=n/2;i++)

{

printf("\nResult of Board %d = ",i);

scanf("%d",&r);

p[r-1].score++;

}

}

void match(struct players p[],int n)

{

int i,board,oboard=0;

i=0;

//n is to store no of players

//board=no of boards

//oboard=no of boards occupied

board=n/2;

printf("\n\_\_\_\_\_\_\_\_\_Pairing for round\_\_\_\_\_\_\_\_\_\_");

printf("\nW\t\tB");

while(oboard<board)

{

if(i<board)

{

printf("\n%d.%-8s VS ",i+1,p[i].nm);

i=i+board;

printf("%d.%-8s",i+1,p[i].nm);

i=i+1;

oboard++;

if(oboard==n/2)

break;

}

if(i>board)

{

printf("\n%d.%-8s VS ",i+1,p[i].nm);

i=i-board;

printf("%d.%-8s",i+1,p[i].nm);

i=i+1;

oboard++;

if(oboard==n/2)

break;

}

}

for(i=0;i<n;i++)

p[i].round++; //for increasing no of round count

}

void standings(struct players p[],int n)

{

int i;

// clrscr();

printf("\_\_\_\_\_Standings\_\_\_\_\_");

printf("\n\tScore");

for(i=0;i<n;i++)

printf("\n%-2d.%-8s\t%d",i+1,p[i].nm,p[i].score);

}

void finalsort(struct players p[],int n)

{

int i,j,srate;

char temp[20];

for(i=0;i<n;i++)

{

for(j=0;j<n-1-i;j++)

{

if((p[j].rate)<(p[j+1].rate))

{

strcpy(temp,p[j+1].nm);

strcpy(p[j+1].nm,p[j].nm);

strcpy(p[j].nm,temp);

srate=p[j+1].rate;

p[j+1].rate=p[j].rate;

p[j].rate=srate;

}

}

}

}

void winners(struct players p[])

{

int i,j,srate;

char temp[20];

printf("\_\_\_\_\_\_\_\_\_\_WINNERS\_\_\_\_\_\_\_\_\_\_\_\_ ");

if((p[0].score!=p[1].score)&&(p[1].score!=p[2].score))

{

printf("\nS.no. Player name Score Rating\n");

printf(" 1.\t %s \t %d \t %d\n",p[0].nm,p[0].score,p[0].rate);

printf(" 2.\t %s \t %d \t %d\n",p[1].nm,p[1].score,p[1].rate);

printf(" 3.\t %s \t %d \t %d\n",p[2].nm,p[2].score,p[2].rate);

}

else

{

if(p[0].score==p[1].score==p[2].score)

{

finalsort(p,3);

printf("\nS.no. Player name Score Rating\n");

printf(" 1.\t %s \t %d \t %d\n",p[0].nm,p[0].score,p[0].rate);

printf(" 2.\t %s \t %d \t %d\n",p[1].nm,p[1].score,p[1].rate);

printf(" 3.\t %s \t %d \t %d\n",p[2].nm,p[2].score,p[2].rate);

}

else if((p[0].score==p[1].score)&&(p[1].score!=p[2].score))

{

finalsort(p,2);

printf("\nS.no. Player name Score Rating\n");

printf(" 1.\t %s \t %d \t %d\n",p[0].nm,p[0].score,p[0].rate);

printf(" 2.\t %s \t %d \t %d\n",p[1].nm,p[1].score,p[1].rate);

printf(" 3.\t %s \t %d \t %d\n",p[2].nm,p[2].score,p[2].rate);

}

else

{

if(p[1].rate<p[2].rate)

{

strcpy(temp,p[1].nm);

strcpy(p[1].nm,p[2].nm);

strcpy(p[2].nm,temp);

srate=p[1].rate;

p[1].rate=p[2].rate;

p[2].rate=srate;

}

printf("\nS.no. Player name Score Rating\n");

printf(" 1.\t %s \t %d \t %d\n",p[0].nm,p[0].score,p[0].rate);

printf(" 2.\t %s \t %d \t %d\n",p[1].nm,p[1].score,p[1].rate);

printf(" 3.\t %s \t %d \t %d\n",p[2].nm,p[2].score,p[2].rate);

}

}

}

void main()

{ int n,i;

int ch;

clrscr();

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_WELCOME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("\nEnter No of players = ");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("\nEnter name & rating of player %d = ",i+1);

scanf("%s",p[i].nm);

scanf("%d",&p[i].rate);

p[i].round=p[i].score=0;

}

clrscr();

sort(p,n);

printf("\n\tRegistered Players are\n");

display(p,n);

label1:

match(p,n);

result(p,n);

sort(p,n);

standings(p,n);

printf("\nDo you want to continue (1/0) = ");

scanf("%d",&ch);

if(ch==1)

{clrscr();

goto label1;

}

else

winners(p);

getch();

}

Output:









