

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define LUNGHEZZA 30
#define MAX_GIOCATORI 10

typedef struct Giocatore_{
    char cognome[LUNGHEZZA+1];
    int punteggio;
    int tutti;
}Giocatore;

void calcolaPunteggio(Giocatore* a, Giocatore* b, int res1, int res2);
int cercaInserisci(Giocatore vec[], char* cognome, int *nG);

int main(int argc, char** argv)
{
    char* gioc = argv[1];

    FILE* fp;
    fp = fopen("partiteD.txt", "r");
    if (fp == NULL) {
        printf("errore! \n");
    }

    Giocatore giocatori[MAX_GIOCATORI];

    char cognome1[LUNGHEZZA+1];
    char cognome2[LUNGHEZZA+1];
    int res1;
    int res2;

    int i;
    int numero_giocatori = 0;
    int posiz1;
    int posiz2;
    int nuovo;
    while(fscanf(fp, "%s %s %d %d", cognome1, cognome2, &res1, &res2) != EOF){
        //posiz1 = cercaInserisci(giocatori, cognome1, &numero_giocatori);
        nuovo = 1;
        for(int i=0; i < numero_giocatori; i++){
            if(strcmp(giocatori[i].cognome, cognome1) == 0){
                giocatori[i].tutti++;
                posiz1 = i;
                nuovo = 0;
            }
        }
        if(nuovo){
            //inizializzazione nuovo elemento
            posiz1 = numero_giocatori;
            strcpy(giocatori[posiz1].cognome, cognome1);
            giocatori[posiz1].punteggio = 0;
            giocatori[posiz1].tutti = 1;
            numero_giocatori++;
        }
        //posiz2 = cercaInserisci(giocatori, cognome2, &numero_giocatori);
        //printf("%d %d \n", posiz1, posiz2);
        nuovo = 1;
        for(int i=0; i < numero_giocatori; i++){
            if(strcmp(giocatori[i].cognome, cognome2) == 0){
                giocatori[i].tutti++;
                posiz2 = i;
                nuovo = 0;
            }
        }
    }
    if(nuovo){

```

```

        //inizializzazione nuovo elemento
        posiz2 = numero_giocatori;
        strcpy(giocatori[posiz2].cognome, cognome2);
        giocatori[posiz2].punteggio = 0;
        giocatori[posiz2].tutti = 1;
        numero_giocatori++;
    }
    //calcolaPunteggio(&giocatori[posiz1],&giocatori[posiz2], res1, res2);
    if( res1 - res2 > 0 ){
        //ha vinto la prima squadra
        giocatori[posiz1].punteggio+= 3;
        giocatori[posiz2].punteggio+= 1;
    }
    else if (res1 - res2 < 0){
        giocatori[posiz1].punteggio+= 1;
        giocatori[posiz2].punteggio+= 3;
    }
    else{
        giocatori[posiz1].punteggio+= 2;
        giocatori[posiz2].punteggio+= 2;
    }
}
fclose(fp);
for(i=0; i<numero_giocatori; i++){
    if(giocatori[i].tutti != numero_giocatori-1){
        printf("Formato non corretto \nMancano partite\n\n");
        return -1;
    }
}
int posiz;
printf("Ci sono %d giocatori\n", numero_giocatori);
for(i=0; i < numero_giocatori; i++){
    //prendo la posizione del giocatore nel vettore di giocatori
    if(strcmp(giocatori[i].cognome, gioc) == 0){
        posiz = i;
        break;
    }
}
int vincitore = 0;
int pareggio = 0;
for(i=0; i < numero_giocatori; i++){
    if(i == posiz)
        continue;
    if(giocatori[posiz].punteggio > giocatori[i].punteggio)
        vincitore = 1;
    if(giocatori[posiz].punteggio < giocatori[i].punteggio){
        vincitore = 0;
        printf("%s non e' un vincitore\n", gioc);
        return 0;
    }
    if(giocatori[posiz].punteggio == giocatori[i].punteggio)
        pareggio = 1;
}
if(vincitore && pareggio){
    printf("%s vincitore parimerito\n", gioc);
}else{
    printf("%s vincitore assoluto\n", gioc);
}
return 0;
}

void calcolaPunteggio(Giocatore *a, Giocatore *b, int res1, int res2){
    if( res1 - res2 > 0 ){
        //ha vinto la prima squadra
        (*a).punteggio+= 3;
        (*b).punteggio+= 1;
    }
}

```

```

    }
    else if (res1 - res2 < 0){
        (*a).punteggio+= 1;
        (*b).punteggio+= 3;
    }
    else{
        (*a).punteggio+= 2;
        (*b).punteggio+= 2;
    }
}

}

int cercaInserisci(Giocatore vec[], char* cognome, int *nG){
    int posiz;
    for(int i=0; i < *nG; i++){
        if(strcmp(vec[i].cognome, cognome) == 0){
            vec[i].tutti++;
            return i;
        }
    }
    //inizializzazione nuovo elemento
    posiz = *nG;
    strcpy(vec[posiz].cognome, cognome);
    vec[posiz].punteggio = 0;
    vec[posiz].tutti = 1;
    (*nG)++;
    return posiz;
}

```