



Esercitazione in C #10

Esercizio 1

Scrivere un programma che legge da file un numero a priori indefinito di numeri interi. Il programma deve richiedere all'utente il nome del file (stringa di massimo 50 caratteri) da cui caricare i dati e se il file non esistesse lo richiede fino a che non viene immesso il nome corretto oppure viene immessa la stringa "fine" che fa terminare il programma. Il programma deve procedere al conteggio delle occorrenze dei numeri presenti nel file, considerando solo quelli compresi tra 0 e 9, estremi compresi, e trascurando i numeri non compresi nell'intervallo. Infine deve stampare l'istogramma di tali occorrenze, per i numeri con almeno una occorrenza.

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

#define MAXLEN 50
#define END_PROGRAM "fine"
#define MAXDIM 10

int main()
{
    FILE *in;
    char fileName[MAXLEN+1];
    int fine, num, occorrenze[MAXDIM];
    int i,j;

    printf("Inserisci il nome del file che vuoi aprire - (fine per terminare):\n");
    gets(fileName);

    fine = 0;
    if (strcmp(fileName, END_PROGRAM) == 0) //!strcmp(fileName, END_PROGRAM)
    {
        printf("Hai terminato il programma\n");
        fine = 1;
    }

    while (!fine) //fine == 0
    {
        in = fopen(fileName, "r");
        if (in == NULL) //Ci sono stati problemi nell'aprire il file
        {
            printf("Errore, file non trovato\n");
            printf("Inserisci il nome del file che vuoi aprire - (fine per terminare):\n");
            gets(fileName);
        }
    }
}
```

```

    if (strcmp(fileName, END_PROGRAM) == 0) //!strcmp(fileName, END_PROGRAM)
    {
        printf("Hai terminato il programma\n");
        fine = 1;
    }
}
else
{
    for (i = 0; i < MAXDIM; i++)
        occorrenze[i] = 0;
    while (!feof(in)&& i < MAXDIM) //feof == 0
    {
        fscanf(in, "%d", &num);
        if (num >= 0 && num <=9)
            occorrenze[num]++;
    }
}
fclose(in);

for(i = 0; i < MAXDIM; i++)
    if (occorrenze[i] > 0)
    {
        printf("\n:%d", i);
        for(j = 0; j < occorrenze[i]; j++)
            printf("*");
    }
    fine = 1; //Programma terminato
}
return 0;
}

```

Esercizio 2

Si scriva un programma che esegue la copia di un file di testo. Il programma legge carattere per carattere il contenuto del file "in.txt" e lo ricopia nel file "out.txt".

```

#include <stdio.h>

int main()
{
    FILE *in, *out;
    char car;
    int letti, scritti;

    in = fopen("in.txt", "r");
    out = fopen("out.txt", "w");

    if (in == NULL || out == NULL)
    {
        if (in == NULL)
            printf("Impossibile aprire il file di input\n");
        else
            fclose(in);
        if (out == NULL)
            printf("Impossibile aprire il file di output\n");
    }
}

```

```

else
    fclose(out);
}
else
{
    letti = fscanf(in, "%c", &car);
    while (letti > 0 && scritti > 0) //Letti non è vuoto
    {
        scritti = fprintf(out, "%c", car);
        letti = fscanf(in, "%c", &car);
    }
    if (ferror(in))
        printf("Errore nel file di input\n");
    if (ferror(out))
        printf("Errore nel file di output\n");
    if (fclose(in) == EOF)
        printf("Errore di chiusura del file di input");
    if (fclose(out) == EOF)
        printf("Errore di chiusura del file di output");
}
return 0;
}

```

Esercizio 3

Scrivere un programma in C per l'acquisizione dei dati e il calcolo dei punteggi per il campionato di Formula 1. Il programma dovrà mostrare all'utente un menu di scelta attraverso il quale potrà:

1. Inserire una nuova scuderia selezionando l'opzione 1. L'utente può inserire un massimo di 10 scuderie. Una scuderia è caratterizzata dalle seguenti informazioni:
 1. Nome, stringa di massimo 50 caratteri;
 2. Pilota 1 e Pilota 2, ognuno caratterizzato dalle seguenti informazioni:
 1. Nome, stringa di massimo 40 caratteri;
 2. Cognome, stringa di massimo 40 caratteri;
 - Elenco delle posizioni di arrivo per i 22 gran premi.

Per acquisire i dati utilizzare una procedura **insDatiScuderia** che accetta come parametro la struttura nella quale memorizzare i valori inseriti dall'utente e acquisisce il nome della scuderia ed i nomi dei piloti.

2. Inserire le posizioni relative ad un gran premio selezionando l'opzione 2. Il programma chiede all'utente il numero del gran premio per il quale si vogliono inserire le posizioni e richiede, per ogni pilota definito, la sua posizione di arrivo. Per acquisire i dati utilizzare una procedura **insDatiGranPremio** che accetta

come parametro il vettore di strutture nel quale memorizzare i valori inseriti dall'utente.

3. Stampare a video i punti effettuati da un pilota. Definire a questo scopo una funzione **puntiPilota** che, ricevuti in ingresso il cognome di un pilota e la struttura che contiene tutti i dati, ritorna i punti effettuati dal pilota secondo la seguente tabella

Tabella

Aa Posizione	# Punti
<u>1</u>	10
<u>2</u>	6
<u>3</u>	4
<u>4</u>	3
<u>5</u>	2
<u>6</u>	1

4. Stampare tutte le informazioni attraverso una procedura denominata **stampa**. La procedura riceve in ingresso tutte le informazioni necessarie e stampa a video tutti i dati.

5. Terminare il programma selezionando l'opzione 0.

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

#define DIM 50
#define NAME 40
#define GP 2
#define MAXSCUD 2

typedef struct pilota
{
    char nome[NAME+1];
    char cognome[NAME+1];
    int posizione[GP];
}pilota;

typedef struct scuderia
{
    char nome[DIM+1];
    pilota pilota1;
    pilota pilota2;
}scuderia;

void mostraMenu();
void insDatiScuderia(scuderia x[]);
void insDatiGranPremio(scuderia x[]);
```

```

int puntiPilota(scuderia x[]);
void stampa(scuderia x[]);
int calcolaPunti(int p[]);

int main()
{
    int scelta, finito, puntifinali;
    finito = 1;
    int i, j;

    scuderia scuderie[MAXSCUD];

    for (i = 0; i < MAXSCUD; i++)
    {
        for (j = 0; j < GP ; j++)
        {
            scuderie[i].pilota1.posizione[j];
            scuderie[i].pilota2.posizione[j];
        }
    }

    while (finito)
    {
        mostraMenu();
        scanf("%d", &scelta);
        getchar();
        switch (scelta)
        {
            case 1:
                insDatiScuderia(scuderie);
                break;
            case 2:
                insDatiGranPremio(scuderie);
                break;
            case 3:
                puntifinali = puntiPilota(scuderie);
                printf("I punti del pilota sono:%d\n", puntifinali);
                break;
            case 4:
                stampa(scuderie);
                break;
            case 0:
                finito = 0;
                break;
            default:
                printf("Attenzione, inserisci un numero compreso tra 0 e 4\n");
        }
    }
    return 0;
}

void mostraMenu()
{
    printf("[1] Inserire una nuova scuderia\n");
    printf("[2] Inserire le posizioni di arrivo di un GP\n");
    printf("[3] Punti realizzati da un pilota\n");
    printf("[4] Visualizza tutti i dati\n");
    printf("[0] Termina il programma\n");
}

```

```

void insDatiScuderia(scuderia x[])
{
    int i;
    for (i = 0; i < MAXSCUD; i++)
    {
        printf("\nNome scuderia:");
        gets(x[i].nome);

        printf("Pilota 1:\n");
        printf("\t Nome:");
        gets(x[i].pilota1.nome);
        printf("\n\t Cognome:");
        gets(x[i].pilota1.cognome);

        printf("\nPilota 2:\n");
        printf("\t Nome:");
        gets(x[i].pilota2.nome);
        printf("\n\t Cognome:");
        gets(x[i].pilota2.cognome);
    }
}

void insDatiGranPremio(scuderia x[])
{
    int ngp;//numero gran premio
    int i;
    do
    {
        printf("Inserisci il numero del gran premio per il quale inserire le posizioni\n");
        scanf("%d", &ngp);
        if (ngp < 1 || ngp > 22)
            printf("ATTENZIONE, i gran premi sono massimo 22\n");
    } while (ngp < 1 || ngp > 22);

    for(i = 0; i < MAXSCUD; i++)
    {
        printf("Scuderia: %s\n", x[i].nome);

        printf("Pilota 1: %s\n", x[i].pilota1.cognome);
        printf("Posizione:");
        scanf("%d", &x[i].pilota1.posizione[ngp-1]);

        printf("\nPilota 2: %s\n", x[i].pilota2.cognome);
        printf("Posizione:");
        scanf("%d", &x[i].pilota2.posizione[ngp-1]);
    }
}

int puntiPilota (scuderia x[])
{
    char cognome[NAME+1];
    int i, punteggio;

    printf("Inserisci il cognome del pilota per vedere i suoi punti\n");
    gets(cognome);

    for (int i = 0; i < MAXSCUD; i++)
    {
        if (strcmp(cognome, x[i].pilota1.cognome) == 0)
        {

```

```

        return calcolaPunti(&x[i].pilota1.posizione[i]);
    }

    if(strcmp(cognome, x[i].pilota2.cognome) == 0)
    {
        return calcolaPunti(&x[i].pilota2.posizione[i]);
    }
}

int calcolaPunti(int p[])
{
    int punteggio = 0, j = 0;
    for (j = 0; j < GP; j++)
    {
        switch (p[j])
        {
            case 1:
                punteggio += 10;
                break;
            case 2:
                punteggio += 6;
                break;
            case 3:
                punteggio += 4;
                break;
            case 4:
                punteggio += 3;
                break;
            case 5:
                punteggio += 2;
                break;
            case 6:
                punteggio += 1;
                break;
            default:
                punteggio += 0;
        }
    }
    return punteggio;
}

void stampa(scuderia x[])
{
    int i;
    for (i = 0; i < MAXSCUD; i++)
    {
        printf("Scuderia:%s\n", x[i].nome);

        printf("Pilota 1:\n");
        printf("\t Nome:%s", x[i].pilota1.nome);
        printf("\n\t Cognome:%s", x[i].pilota1.cognome);
        printf("Punteggio: %d\n", calcolaPunti(&x[i].pilota1.posizione[i]));

        printf("\nPilota 2:\n");
        printf("\t Nome:%s", x[i].pilota2.nome);
        printf("\n\t Cognome:%s", x[i].pilota2.cognome);
        printf("Punteggio: %d\n", calcolaPunti(&x[i].pilota2.posizione[i]));
    }
}

```

```
}  
}
```



THEUNINOTES.COM