

ESERCIZIO 1

Si realizzi un programma C che legga da utente i dati relativi ad alcuni corsi. In particolare, per ogni corso vengono dati:

- **denominazione** del corso: una stringa di 20 caratteri che riporta il nome del corso;
- **cognome** del docente: una stringa di 15 caratteri che rappresenta il cognome del docente del corso;
- **iscritti**: un intero che indica il numero di studenti che frequentano il corso.

Il programma deve stampare la denominazione del corso e il cognome del docente relativi a tutti i corsi che hanno il numero di iscritti maggiore o uguale alla media aritmetica degli iscritti (calcolata su tutti i corsi).

ESERCIZIO 1

Attenzione: abbiamo bisogno di un ARRAY di strutture !!!!!

Esempio: l'utente inserisce i seguenti dati per 3 corsi

analisi

dore

55

fond.inf

milano

40

geometria

gilotti

37

| | | |
|----------------|-----------------|------------------|
| <i>analisi</i> | <i>fond.inf</i> | <i>geometria</i> |
| <i>dore</i> | <i>milano</i> | <i>gilotti</i> |
| <i>55</i> | <i>40</i> | <i>37</i> |

La media e' di 44 quindi il programma stampa:

analisi

dore

ESERCIZIO 1

```
#include <stdio.h>
#define N 30
struct stud {
    char denominazione[21];
    char cognome_docente[16];
    int studenti;
} corsi[N];

void main()
{
    int i, nc;
    float somma, media;
    printf("Inserisci il numero dei corsi ");
    scanf("%d", &nc);
    /* inserimento dati */
    for (i=0; i<nc; i++)
    {
        printf("Inserisci il nome del corso : ");
        scanf("%s",corsi[i].denominazione);
        printf("Inserisci il cognome del docente : ");
        scanf("%s",corsi[i].cognome_docente);
        printf("Inserisci il numero degli iscritti : ");
        scanf("%d",&corsi[i].studenti);
    }
}
```

Continua...

ESERCIZIO 1

```
somma=0;
for (i=0; i< nc; i++)
    somma=somma + corsi[i].studenti;
media= somma/nc;
for (i=0; i< nc; i++)
    if (corsi[i].studenti>=media)
        printf("%s\t%s\n",corsi[i].denominazione,
            corsi[i].cognome_docente);
}
```

ESERCIZIO 2

1) Si scriva un programma C che legga una serie di dati e li memorizzi in un vettore SQUADRE (di dimensione 3) contenente strutture (**struct squadra**) del tipo:

- nome squadra** (stringa di lunghezza 20)
- codice squadra** (intero)
- goal fatti** (intero)
- goal subiti** (intero)

2) Stampi a terminale tutti i nomi e codici delle squadre che hanno fatto un numero di goal maggiore del numero dei goal subiti.

3) Letto a terminale un codice di una squadra stampi a video il nome della squadra, i goal fatti e i goal subiti.

ESERCIZIO 2

Attenzione: abbiamo bisogno di un ARRAY di strutture !!!!!

Esempio: l'utente inserisce i seguenti dati per 3 squadre

juventus

1

26

6

sampdoria

2

23

17

lecce

3

8

21

| <i>juventus</i> | <i>sampdoria</i> | <i>lecce</i> |
|-----------------|------------------|--------------|
| <i>1</i> | <i>2</i> | <i>3</i> |
| <i>26</i> | <i>23</i> | <i>8</i> |
| <i>6</i> | <i>17</i> | <i>21</i> |

2) Viene stampato a video

juventus 1

sampdoria 2

3) Se l'utente digita 3 viene stampato

lecce 8 21

ESERCIZIO 2

```
#include <stdio.h>
#define N 30

void main()
{
    struct squadra{
        char nome[21];
        int codice;
        int goal_fatti, goal_subiti;
    } SQUADRE[N];

    int i, ns, cod, T;

    printf("Inserisci il numero delle squadre");
    scanf("%d", &ns);
    /* inserimento dati */
    for (i=0; i<ns; i++)
    {
        printf("Inserisci nome, codice, goal fatti e subiti \n");
        scanf("%s",SQUADRE[i].nome);
        scanf("%d",&SQUADRE[i].codice);
        scanf("%d",&SQUADRE[i].goal_fatti);
        scanf("%d",&SQUADRE[i].goal_subiti);
    }
}
```

ESERCIZIO 2

```
/* punto 2 */
for (i=0; i<ns; i++)
    { if(SQUADRE[i].goal_fatti> SQUADRE[i].goal_subiti)
        {printf("%s\n",SQUADRE[i].nome);
         printf("%d\n",SQUADRE[i].codice);
        }
    }

/* punto 3 */
printf("Inserisci un codice ");
scanf("%d", &cod);
i=0; T=0;
while ((i < ns)&& (T==0))
    { if (SQUADRE[i].codice == cod)
        {printf("%s\n",SQUADRE[i].nome);
         printf("%d\n",SQUADRE[i].goal_fatti);
         printf("%d\n",SQUADRE[i].goal_subiti);
         T=1;}
        i++;
    }
if (T==0) printf("codice non trovato");
}
```

ESERCIZIO 3

- 1) Scrivere una procedura che copi in una stringa destinazione una parte di una stringa sorgente, dal carattere alla posizione n1 al carattere alla posizione n2.
- 2) Scrivere un main cliente di prova.

Osservazione: non viene fatto alcun controllo sugli estremi della sottostringa, né sulla dimensione della stringa destinazione. Modificare il programma inserendo tali controlli.

ESERCIZIO 3

```
#include <stdio.h>

void sottostr(char sorg[], char dest[], int n1, int n2)
{
    int i=0;
    while ((sorg[n1]!='\0') && (n1<=n2))
    {
        dest[i]=sorg[n1];
        i++; n1++;
    }
    dest[i]='\0';
}

void main()
{
    char sorg[80], dest[80];
    int n1,n2;
    printf("Inserire una stringa: ");
    scanf("%s",sorg);
    printf("Inizio sottostringa: ");
    scanf("%d",&n1);
    printf("Fine sottostringa: ");
    scanf("%d",&n2);
    sottostr(sorg,dest,n1,n2);
    printf("La sottostringa e' %s", dest);
}
```

ESERCIZIO 3

Main con il controllo dei valori

```
void main()
{char sorg[80], dest[80];
  int n1,n2;
  printf("Inserire una stringa: ");
  scanf("%s",sorg);
  do
  {printf("Inizio sottostringa: ");
    scanf("%d",&n1);
    printf("Fine sottostringa: ");
    scanf("%d",&n2);
  }
  while ((n1>n2)|| (n2>strlen(sorg)));
  sottostr(sorg,dest,n1,n2);
  printf("La sottostringa e' %s", dest);
}
```