

Esercizi di preparazione al compito:
PROVA PRATICA

Si scriva un programma C che tramite tre funzioni, *leggi*, *media*, *stampa*:

- a) Legga da terminale una prima sequenza di numeri terminati dal valore 0 (un numero su ogni linea) e li inserisca in un vettore A;
- b) Legga da terminale una seconda sequenza di numeri terminati dal valore 0 e li inserisca in un altro vettore B;
- c) Sia in grado di calcolare la media degli elementi di un vettore con la funzione *media*;
- d) Stampi a video il vettore (A oppure B) la cui media è maggiore.

NOTA: Si ipotizzi una dimensione massima di 10 per i vettori

Esempio:

Vettore A:	3	5	7	8	2		Media=25/5=5
Vettore B:	2	6	10	2	3	15	Media=38/6=6.333
Vettore Max:	2	6	10	2	3	15	

Soluzione:

```
#include<stdio.h>
#define MAX 10

int leggi(int vet[],char nome);
float media(int vet[],int lung);
void stampa(int vet[],int lung);

main()
{int A[MAX],B[MAX],a,b;
a=leggi(A,'A');
b=leggi(B,'B');
if(media(A,a)<media(B,b))stampa(B,b);
    else if(media(A,a)==media(B,b))printf("I vettori hanno la stessa
media!\n");
    else stampa(A,a);
}

int leggi(int vet[],char nome)
{int i=0;
printf("\nScrivi gli elementi del vettore %c\n",nome);
do
    {printf("Elemento %d: ",i+1);
    scanf("%d",&vet[i]);
    i++;
    }
while(vet[i-1]!=0&&i<MAX);
return i-1;
}
```

```
float media(int vet[],int lung)
{int i;
float sum=0;
for(i=0;i<lung;i++)sum+=vet[i];
return sum/lung;
}

void stampa(int vet[],int lung)
{int i;
printf("Il vettore con media maggiore è:\n");
for(i=0;i<lung;i++)printf("%d ",vet[i]);
}
```

Esercizio

Si scriva un programma in linguaggio C che tramite:

- la funzione *leggi*, legga a terminale i dati di N studenti (con N definito come la costante 4) costituiti da Nome, Cognome, Voto, e li inserisca in un vettore;
- calcoli con la funzione *media* la media dei voti;
- stampi a terminale i nominativi di ciascuno studente
- stampi poi a terminale il voto medio.

Esempio:

Avendo in ingresso:

Alberani
Luigi
30
Vettori
Piero
33
Zanetti
Lorenzo
18

Stampa:

Alberani
Luigi
Vettori
Piero
Zanetti
Lorenzo

Voto medio: 27.0

Soluzione:

```
#include <stdio.h>
#define N 4

struct studente{char Nome[20];
                 char Cognome[20];
                 int Voto;};

void leggi(int n, struct studente Vet[]);
float media(int n, struct studente Vet[]);

main()
{int i;
 struct studente V[N];
 leggi(N,V);
 for(i=0;i<N;i++)
 {printf("%s\n",V[i].Nome);
  printf("%s\n", V[i].Cognome); }
 printf("\nVoto medio: %f\n",media(N,V)); }

void leggi(int n, struct studente Vet[])
{int i,j;
 for(i=0;i<n;i++)
 {printf("Inserisci Nome, Cognome e Voto");
  scanf("%s",Vet[i].Nome);
  scanf("%s",Vet[i].Cognome);
  scanf("%d",&Vet[i].Voto);}
}

float media(int n, struct studente Vet[])
{int i;
 float m=0.0;
 for(i=0;i<n;i++)
 {m=m+Vet[i].Voto;}
 return m/n;
}
```

Esercizio

Si scriva un programma in linguaggio C che tramite:

- la funzione *leggi*, legga a terminale i dati di N corsi (con N definito come la costante 3) costituiti da NomeCorso, NomeDocente, VotoMedio, e li inserisca in un vettore V;
- determini, con la funzione *piu_facile*, quale tra i corsi ha il voto medio più alto e stampi il nome di tale corso a terminale.

Esempio:

Avendo in ingresso:

```
Informatica
Alberani
25
Disegno
Grandi
27
Matematica
Zanetti
18
```

Stampa:

```
Disegno
```

Soluzione:

```
#include <stdio.h>
#define N 3

struct studente {char Materia[20];
                  char CognomeDoc[20];
                  int Voto;};

void leggi(int n, struct studente Vet[]);
int piufacile(int n, struct studente Vet[]);

main()
{int i;
 struct studente V[N];
 leggi(N,V);
 i=piufacile(N,V);
 printf("\nPiu' facile: %s\n",V[i].Materia);
}
```

```

void leggi(int n, struct studente Vet[])
{int i,j;
  for(i=0;i<n;i++)
    {printf("Inserisci Materia, Cognome e Voto");
      scanf("%s",Vet[i].Materia);
      scanf("%s",Vet[i].CognomeDoc);
      scanf("%d",&Vet[i].Voto);}
}

int piufacile(int n, struct studente Vet[])
{int i, imax;
  /* imax rappresenta l'indice dell'elemento piu' grande */
  imax=0;
  for(i=0;i<n;i++)
    {if((Vet[i].Voto)>Vet[imax].Voto) imax=i; }
  return imax;
}

```

Esercizio

Si scriva un programma C che legga due serie di dati e li memorizzi in due vettori di strutture. Nel primo vettore M (di dimensione 3) vengono memorizzati dati del tipo:

- matricola
- nome
- cognome

Si noti che la matricola identifica univocamente uno studente e che non ci sono due strutture che contengono lo stesso numero di matricola.

Nel secondo vettore V (di dimensione 7) vengono memorizzati dati del tipo:

- matricola
- esame
- voto

Possono esserci piu' record con lo stesso numero di matricola che denotano diversi esami fatti dallo stesso studente.

Si scriva un programma che tramite opportune procedure legga i dati in ingresso e li inserisca nei due vettori. Successivamente per ogni studente con matricola X contenuto nel vettore M sommi tutti i suoi voti ottenuti negli esami contenuti nel vettore V.


```

#include <stdio.h>
#define DIMN 3
#define DIMM 7

struct studente{
    int matricola;
    char Nome[20];
    char Cognome[20];};

struct esami{
    int matricola;
    char esame[20];
    int voto;};

void leggistud(int n, struct studente V[]);
void leggiesami(int n, struct esami M[]);
int somma(struct studente s, struct esami E[]);

main()
{int i, S=0;
 struct studente V[DIMN];
 struct esami M[DIMM];
 leggistud(DIMN,V);
 leggiesami(DIMM,M);

 for(i=0; i< DIMN;i++)
 {S= somma(V[i], M);
  printf("Studente %s %s\n", V[i].Cognome, V[i].Nome);
  printf("Somma voti: %d\n",S);
 }
}

void leggistud(int n, struct studente Vet[])
{int i,j;
 for(i=0;i<n;i++)
 {printf("Inserisci Matricola, Nome, Cognome\n");
  scanf("%d",&Vet[i].matricola);
  scanf("%s",Vet[i].Nome);
  scanf("%s",Vet[i].Cognome);
 }
}

void leggiesami(int n, struct esami Vet[])
{int i;
 for(i=0;i<n;i++)
 {printf("Inserisci Matricola, Esame, Voto\n");
  scanf("%d",&Vet[i].matricola);
  scanf("%s",Vet[i].esame);
  scanf("%d",&Vet[i].voto);
 }
}

```

```
int somma(struct studente s, struct esami E[])
{int i;
 int S=0;
 for(i=0;i < DIMM;i++)
     if (E[i].matricola == s.matricola)
         S = S + E[i].voto;
 return S;
}
```