

ESERCIZIO 1

Si scriva un programma che

- 1) richieda all'utente un valore V di soglia
- 2) successivamente prenda in ingresso una sequenza di reali positivi terminata da 0 (massimo 10), e memorizzi in un vettore di float M (di dimensione fisica 10) SOLO i valori maggiori di V.
- 3) Infine, il programma deve creare un secondo vettore MED in cui l'elemento i-esimo e' calcolato come la media tra l'elemento i-esimo del vettore M e il valore V.

Per effettuare la media tra due valori si usi la funzione
`float media(float X, float Y);`

ESERCIZIO 1

Esempio: l'utente inserisce il valore 2.5 di soglia.

Poi inserisce la sequenza

1.3 4 5.2 9.5 2.2 1 0

Nel vettore M vengono quindi memorizzati solo

| | | | |
|---|---|-----|-----|
| | 0 | 1 | 2 |
| M | 4 | 5.2 | 9.5 |

Infine, il programma deve creare un secondo vettore MED in cui l'elemento i-esimo e' calcolato come la media tra l'elemento i-esimo del vettore M e il valore V.

| | | | |
|-----|------|------|-----|
| | 0 | 1 | 2 |
| MED | 3.25 | 3.85 | 6.0 |

ESERCIZIO 1

```
float media(float X, float Y){
    return (X+Y)/2;
}
void main()
{float V, num; int i=0, j=0;
 float M[10], MED[10];
 printf("Inserisci la soglia : ");
 scanf("%f",&V);
 do {
     printf("Inserisci elemento : ");
     scanf("%f",&num);
     if (num > V) {M[j]=num; j++;}
 }
 while ((num !=0)&&(++i<10)); /* i=dimensione logica*/
 for(i=0; i<j; i++)
     {MED[i]=media(M[i],V);
      printf("%f\n", MED[i]);}
 }
```

ESERCIZIO 2

Si scriva un programma che prenda in ingresso una sequenza di massimo 10 reali positivi terminata da 0, e la memorizzi in un vettore di float **NUM**.

Il programma deve creare un secondo vettore **MEDIE** in cui l'elemento i-esimo e' calcolato come la media tra l'elemento i-esimo del vettore **NUM** e il suo successivo. Ovviamente la dimensione logica di medie sarà pari alla dimensione logica di **NUM** meno 1.

MEDIE[i] = media(NUM[i], NUM[i+1])

Per effettuare la media si usi la procedura

void media(float X, float Y, float *med);

ESERCIZIO 2

Esempio: l'utente inserisce la sequenza

1.3 4 5.2 9.5 2.2 1 0

| | | | | | | | |
|-----|-----|---|-----|-----|-----|---|---|
| NUM | 1.3 | 4 | 5.2 | 9.5 | 2.2 | 1 | 0 |
|-----|-----|---|-----|-----|-----|---|---|

`MEDIE[i] = media(NUM[i], NUM[i+1])`

| | | | | | | |
|-------|------|------|------|------|-----|-----|
| MEDIE | 2.65 | 4.60 | 7.35 | 5.85 | 1.6 | 0.5 |
|-------|------|------|------|------|-----|-----|

ESERCIZIO 2

```
#include <stdio.h>

void media(float X, float Y, float *med)
{
    *med=(X+Y)/2;
}

void main()
{
    int i=0, j=0;
    float NUM[10], MED[10];
    printf("Inserisci elemento : ");
    scanf("%f",&NUM[i]);
    while ((NUM[i]!=0)&&(++i<10)) /* i=dimensione logica*/
    {
        printf("Inserisci elemento : ");
        scanf("%f",&NUM[i]);
    }
    for(j=0; j<i-1; j++)
    {
        media(NUM[j+1],NUM[j],&MED[j]);
        printf("%f\n", MED[j]);
    }
}
```

ESERCIZIO 3

Scrivere un programma che

- 1) prenda in ingresso due vettori V1 e V2 di interi di dimensione N
- 2) costruisca un terzo vettore V3 di dimensione 2N i cui elementi di posizione pari siano gli elementi del primo vettore e gli elementi di posizione dispari siano gli elementi del secondo vettore.

| | | | | | | |
|----|---|---|---|---|---|---|
| v1 | 4 | 5 | 9 | | | |
| v2 | 2 | 6 | 1 | | | |
| v3 | 0 | 1 | 2 | 3 | 4 | 5 |
| | 4 | 2 | 5 | 6 | 9 | 1 |

ESERCIZIO 3

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

void main()
{int i, v1[N],v2[N],v3[2*N];

for (i=0; i<N; i++)
{   printf("Inserisci elemento %d del primo vettore : ", i);
    scanf("%d",&v1[i]);
}
for (i=0; i<N; i++)
{   printf("Inserisci elemento %d del secondo vettore : ", i);
    scanf("%d",&v2[i]);
}
for (i=0; i<N; i++)
{   v3[2*i] = v1[i];
    v3[2*i+1] = v2[i];
}
for (i=0; i<2*N; i++)
    printf("%d ", v3[i]);
}
```

ESERCIZIO 4

Scrivere un programma che, dato un vettore **NUM** di **N** interi positivi inseriti dall'utente, ne produca due **PAR** e **DIS** contenenti, rispettivamente, i numeri pari e dispari del vettore iniziale.

Si controlli che i numeri inseriti dall'utente siano positivi.

Per verificare se un numero e' pari, si scriva una funzione che restituisce 1 se il numero e' pari, 0 altrimenti

```
int pari(int n);
```

ESERCIZIO 4

Esempio: l'utente inserisce la sequenza

4 2 -4 5 6 -9 1 6
 ↙ scartati ↘

NUM

| | | | | | |
|---|---|---|---|---|---|
| 4 | 2 | 5 | 6 | 1 | 6 |
|---|---|---|---|---|---|

PAR

| | | | | | |
|---|---|---|---|--|--|
| 4 | 2 | 6 | 6 | | |
|---|---|---|---|--|--|

DIMENSIONE LOGICA 4

DIS

| | | | | | |
|---|---|--|--|--|--|
| 5 | 1 | | | | |
|---|---|--|--|--|--|

DIMENSIONE LOGICA 2

ESERCIZIO 4

```
#include <stdio.h>
#define N 20

int pari(int n){
    return (n%2) ? 0 : 1;
}

void main()
{int num[N], par[N], dis[N];
  int i,ivp=0,ivd=0; /* ivp = indice vett pari e ivd = indice vett
dispari*/

  for (i=0; i<N; i++) /*lettura vettore num */
    do {printf("inserire intero positivo del vettore NUM ");
        scanf("%d", &num[i]);
        }
        while (num[i] < 0);
```

Continua...

ESERCIZIO 4

```
for (i=0; i<N; i++) /* trasferimento pari in PAR e dispari in DIS*/
  if (pari(num[i]))
    par[ivp++] = num[i];
  else dis[ivd++] = num[i];

/* stampa dei vettori PAR e DIS
  ATTENZIONE !!!!! La dimensione logica di PAR e' ivp mentre
                    la dimensione logica di DIS e' ivd */

for (i=0; i<ivp; i++)
  printf("%d ", par[i]);

printf("\n");

for (i=0; i<ivd; i++)
  printf("%d ", dis[i]);

}
```