

# Esercizio 1 - cicli

---

## Sequenze di cifre

- Realizzare un programma che prende in input una sequenza di cifre (tra 1 e 9) e calcola la somma massima fra le sotto-sequenze di cifre non decrescente
- Il programma termina quando viene inserito lo 0
- Esempio:

2	2	4	5	3	9	3	1	5	0
	13			12		3	6		

1

## Esercizio 1 – cicli - Soluzione

---

```
#include <stdio.h>
int main() {
    char cifra;
    int cur_val = 0, old_val = 0, somma = 0, somma_max = 0;
    do {
        scanf("%c", &cifra);
        getchar();
        cur_val = cifra - '0';
        if(cur_val < old_val) {
            if( somma > somma_max)
                somma_max = somma;
            somma = cur_val;
        }
        else
            somma = somma + cur_val;
        old_val = cur_val;
    } while(cur_val != 0);
    printf("Massima somma trovata: %d\n", somma_max);
    return 0;
}
```

2

## ESERCIZIO 2

---

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 crei un secondo vettore **MED** in cui l'elemento *i*-esimo è calcolato come la media tra l'elemento *i*-esimo del vettore M e il valore V.

## ESERCIZIO 2

---

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 2 - Soluzione

---

```
#include <stdio.h>

int main() {
    float V, num;
    float M[10], MED[10];
    int i=0, j=0;

    printf("Inserisci la soglia");
    scanf("%f",&V);
    do {
        printf("Inserisci elemento");
        scanf("%f",&num);
        if (num > V) {
            M[i]=num;
            i++;
        }
    } while ( (num!=0) && (i<10) ); // i=dimensione logica

    for(j=0; j<i; j++) {
        MED[j] = (M[j] + V)/2;
        printf("%f\n", MED[j]);
    }
    return 0;
}
```

## ESERCIZIO 3

---

Si scriva un programma che prende 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.

## ESERCIZIO 3

---

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	2.65	4.60	7.35	5.85	1.6	0.5
-------	------	------	------	------	-----	-----

## ESERCIZIO 3 - Soluzione

---

```
#include <stdio.h>

int main() {
    int i=0, j=0;
    float NUM[10], MED[10];

    do {
        printf("Inserisci elemento");
        scanf("%f",&NUM[i]);
        i++;
    } while ( (NUM[i-1]!=0) && (i<10) );

    for(j=0; j<i-1; j++) {
        MED[j] = (NUM[j+1] + NUM[j])/2;
        printf("%f\n", MED[j]);
    }
    return 0;
}
```

## ESERCIZIO 4

---

Scrivere un programma che

1) Legga da input 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.

<b>V1</b>	<table border="1"><tr><td>4</td><td>5</td><td>9</td></tr></table>	4	5	9	<b>V3</b>	<table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>4</td><td>2</td><td>5</td><td>6</td><td>9</td><td>1</td></tr></table>	0	1	2	3	4	5	4	2	5	6	9	1
4	5	9																
0	1	2	3	4	5													
4	2	5	6	9	1													
<b>V2</b>	<table border="1"><tr><td>2</td><td>6</td><td>1</td></tr></table>	2	6	1														
2	6	1																

## ESERCIZIO 4 - Soluzione

---

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

int 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]);
    return 0; }
```

## ESERCIZIO 5

---

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.

## ESERCIZIO 5

---

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 5 - Soluzione

---

```
#include <stdio.h>
#define N 8

int pari(int n){
    if (n%2 == 0) return 1;
    else return 0;
}

int 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 5 - Soluzione

---

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

/* 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]);
return 0; }
```

## Esercizio 6 - Media e Deviazione Standard

---

- Realizzare un programma che, letto un array di interi (sequenza terminata da 0), sia in grado di effettuare il calcolo della media, ed il calcolo della deviazione standard

- Date N misure della stessa grandezza x

- La media è definita come:  $\bar{x} = \frac{\sum_{i=1}^N x_i}{N}$

- La dev. standard è definita come:  $\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}}$

## Esercizio 6 - Media e Deviazione Standard

---

- Come si calcola la radice quadrata di un numero?
- Il C mette a disposizione una “libreria” di funzioni matematiche...
- Per poterle utilizzare, basta:

1. Aggiungere all’inizio del file, la direttiva

```
#include <math.h>
```

1. La radice quadrata di un numero X si calcola con la seguente istruzione:

```
radice = sqrt(X);
```

## Esercizio 6 - Media e Deviazione Standard

---

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

int main() {
    int values[DIM], size, i;
    float media, deviazione, sum;
    size = 0;
    do {
        printf("Inserisci numero: ");
        scanf("%d", &values[size]);
        size++;
    } while (values[size-1]!=0 && size<DIM);

    ...
```

*Continua...*

## Esercizio 6 - Media e Deviazione Standard

---

```
...

    sum = 0;
    media = 0;
    for (i=0; i<size; i++)
        sum = sum + values[i];
    media = sum / size;

    sum = 0;
    for (i=0; i<size; i++)
        sum = sum + (values[i] - media)*(values[i] - media);
    deviazione = sqrt(sum / size);

    printf("Media: %f\n", media);
    printf("Deviazione: %f\n", deviazione);

    return 0;
}
```