

## CS0424IT — ESERCITAZIONE S2 L3

*Simone La Porta*



---

### TRACCIA

*Realizzare due programmi in C:*

- 1. un programma che esegua l'operazione di moltiplicazione tra due numeri inseriti dall'utente;*
- 2. un programma che legga due valori interi e visualizzi la loro media aritmetica.*

### SVOLGIMENTO

#### *Script moltiplicazione*

Programma C che calcola il prodotto di una serie di numeri in virgola mobile inseriti dall'utente.

- Il programma inizia includendo la libreria standard `stdio.h` e dichiarando la funzione `main`. Il prodotto è inizializzato a 1.0 in quanto la moltiplicazione con 1 non altera il risultato.
- Il ciclo `do-while` permette all'utente di inserire numeri in virgola mobile. La funzione `scanf` legge il numero inserito e verifica se l'input è valido. Se l'input non è valido, viene visualizzato un messaggio di errore e il buffer di input viene svuotato per prevenire comportamenti imprevisti.

```
18 #include <stdio.h>
17
16 int main() {
15     float product = 1.0f;
14     char choice;
13
12     do {
11         float number;
10
9         printf("Inserisci un numero in virgola mobile (o 'q' per uscire): ");
8         if (scanf("%f", &number) != 1) {
7             fprintf(
6                 stderr,
5                 "Errore: Input non valido. Per favore inserisci un numero o 'q'.\\n");
4                 while ((getchar()) != '\\n')
3                     ;
2                 continue;
1             }
19         if (number == 0.0f) {
1             printf("Attenzione: Moltiplicando per 0 si otterrà 0.\\n");
2             }
3         product *= number;
4         printf("Vuoi inserire un altro numero (y/n)? ");
5         scanf(" %c", &choice);
6     } while (choice == 'y' || choice == 'Y');
7     printf("Il prodotto di tutti i numeri inseriti è: %.5f\\n", product);
8
9     return 0;
10 }
```

NORMAL mu.c utf-8 65% 19:25  
"mu.c" 29L, 721B written

Figura 1: Programma moltiplicazione.

- Se l'utente inserisce il numero 0, viene visualizzato un avviso. Il numero inserito viene quindi moltiplicato per il prodotto corrente. Dopo di che, viene chiesto all'utente se desidera inserire un altro numero. Il ciclo continua finché l'utente risponde con 'y' o 'Y'.
- Alla fine, il programma visualizza il prodotto di tutti i numeri inseriti con una precisione di cinque cifre decimali.

---

## *Script media*

Programma C che calcola la media, la mediana e la deviazione standard di un insieme di numeri forniti dall'utente.

- **Definizioni e funzioni di supporto:**

- Il codice inizia con l'inclusione delle librerie standard `stdio.h` per l'input/output, `math.h` per le funzioni matematiche e `stdbool.h` per il supporto ai tipi booleani.
- Si definisce una costante `MAX_NUMBERS` che indica il numero massimo di numeri che l'utente può inserire (100).
- La funzione `get_float` visualizza un prompt all'utente e legge un valore float. Restituisce `true` se l'input è valido. Questa funzione prende due argomenti: un prompt da visualizzare e un puntatore a `float` dove memorizzare il valore letto.
- La funzione `get_confirmation` chiede all'utente una conferma (sì/no) e restituisce `true` se la risposta è affermativa (y o Y). Questa funzione prende un prompt da visualizzare come argomento.

- **Calcolo delle statistiche:**

- La funzione `calculate_statistics` calcola la media, la mediana e la deviazione standard di un array di numeri. Se il numero di elementi è zero, la funzione restituisce immediatamente. Per calcolare la media, somma tutti i numeri e divide per il numero di elementi.
- Per calcolare la mediana, l'array dei numeri viene ordinato utilizzando il bubble sort (per semplicità). La variabile `swapped` tiene traccia se ci sono stati scambi durante l'ordinamento.
- La mediana viene calcolata in base al numero di elementi: se è pari, la mediana è la media dei due elementi centrali, altrimenti è il valore centrale.
- La deviazione standard viene calcolata sommando i quadrati delle differenze tra ogni numero e la media, quindi calcolando la radice quadrata della somma divisa per il numero di elementi.

- **Funzione principale:**

- 
- La funzione `main` gestisce l'interazione con l'utente per ottenere i numeri. Visualizza un messaggio di benvenuto, poi in un ciclo chiede all'utente di inserire numeri fino a quando l'utente decide di smettere o si raggiunge il numero massimo di input.
  - Se non sono stati inseriti numeri, il programma termina con un messaggio appropriato. Altrimenti, calcola e stampa le statistiche (media, mediana, deviazione standard) utilizzando la funzione `calculate_statistics`.

```

1  #include <math.h>
2  #include <stdbool.h>
3  #include <stdio.h>
4  #define MAX_NUMBERS 100 // Numero massimo di input consentiti (100)
5  bool get_float(const char *prompt, float *value) {
6      printf("\n%s", prompt);
7      return scanf("%f", value) == 1;
8  }
9
10 bool get_confirmation(const char *prompt) {
11     char answer;
12     printf("%s (y/n): ", prompt);
13     return scanf(" %c", &answer) == 1 && (answer == 'y' || answer == 'Y');
14 }
15
16 void calculate_statistics(float numbers[], int num_elements, float *mean,
17                          float *median, float *standard_deviation) {
18     if (num_elements == 0) {
19         return;
20     }
21
22     *mean = 0.0f;
23     for (int i = 0; i < num_elements; i++) {
24         *mean += numbers[i];
25     }
26     *mean /= num_elements;
27
28     bool swapped;
29     do {
30         swapped = false;
31         for (int i = 0; i < num_elements - 1; i++) {
32             if (numbers[i] > numbers[i + 1]) {
33                 float temp = numbers[i];
34                 numbers[i] = numbers[i + 1];
35                 numbers[i + 1] = temp;
36                 swapped = true;
37             }
38         }
39     } while (swapped);
40
41     if (num_elements % 2 == 0) {
42         *median =
43             (numbers[num_elements / 2 - 1] + numbers[num_elements / 2]) / 2.0f;
44     } else {
45         *median = numbers[num_elements / 2];
46     }
47
48     *standard_deviation = 0.0f;
49     for (int i = 0; i < num_elements; i++) {
50         float deviation = numbers[i] - *mean;
51         *standard_deviation += deviation * deviation;
52     }
53     *standard_deviation = sqrt(*standard_deviation / num_elements);
54 }
55
56 int main() {

```

Figura 2: Programma media pt.1.

```

9 }
8
7 *standard_deviation = 0.0f;
6 for (int i = 0; i < num_elements; i++) {
5     float deviation = numbers[i] - *mean;
4     *standard_deviation += deviation * deviation;
3 }
2 *standard_deviation = sqrt(*standard_deviation / num_elements);
1 }
56
1 int main() {
2     float numbers[MAX_NUMBERS];
3     int num_elements = 0;
4
5     printf("Questo programma calcola la media, la mediana e la deviazione "
6           "standard di un insieme di numeri.\n");
7
8     do {
9         float number;
10        if (!get_float("Inserisci un numero (o 'q' per uscire): ", &number)) {
11            fprintf(stderr, "Errore: Input non valido.\n");
12            continue;
13        }
14
15        if (num_elements == MAX_NUMBERS) {
16            fprintf(stderr,
17                    "Attenzione: Numero massimo di elementi raggiunto (%d).\n",
18                    MAX_NUMBERS);
19            break;
20        }
21
22        numbers[num_elements++] = number;
23    } while (get_confirmation("Aggiungere un altro numero?"));
24
25    if (num_elements == 0) {
26        printf("Nessun numero inserito.\n");
27        return 0;
28    }
29
30    float mean, median, standard_deviation;
31    calculate_statistics(numbers, num_elements, &mean, &median,
32                        &standard_deviation);
33
34    printf("\nStatistiche:\n");
35    printf("  Numero di input:      %d\n", num_elements);
36    printf("  Media:                  %.5f\n", mean);
37    printf("  Mediana:               %.5f\n", median);
38    printf("  Deviazione standard:   %.5f\n", standard_deviation);
39
40    return 0;
41 }

```

NORMAL me.c [+]

10 < utf-8 < 57% 56:1

Figura 3: Programma media pt.2.