

## Esercizi S2-L3

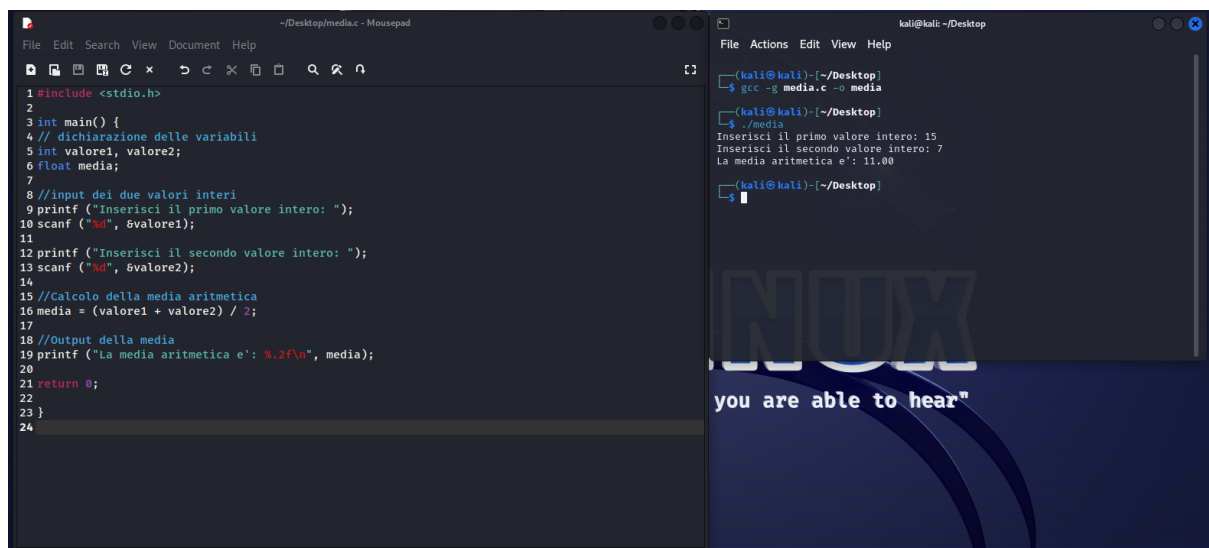
Cosa significa programmare?

Programmare significa un algoritmo in linguaggio di programmazione che poi verrà convertito in linguaggio macchina dal computer. Questo permetterà alla macchina di interpretarlo. Un algoritmo è una sequenza di regole che vengono usate per risolvere un problema specifico. Dovrà rispettare 3 regole:

- Deve essere non ambiguo: le istruzioni devono essere ben definite
- Deve essere finito: ci deve essere una situazione che fa finire l'algoritmo
- Deve essere eseguibile: una volta trascritto dovrà essere proponibile nella realtà

Il linguaggio può essere di due tipologie:

- Compilato: Il codice deve essere completamente eseguito e salvato (linguaggio C è di tipo compilato)
- Interpretato: il codice viene eseguito senza che esso venga installato (esempio: Script).



The image shows a code editor window titled '~Desktop/media.c - Mousepad' and a terminal window titled 'kali@kali:~/Desktop'. The code editor contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     // dichiarazione delle variabili
5     int valore1, valore2;
6     float media;
7
8     //input dei due valori interi
9     printf ("Inserisci il primo valore intero: ");
10    scanf ("%d", &valore1);
11
12    printf ("Inserisci il secondo valore intero: ");
13    scanf ("%d", &valore2);
14
15    //Calcolo della media aritmetica
16    media = (valore1 + valore2) / 2;
17
18    //Output della media
19    printf ("La media aritmetica e': %.2f\n", media);
20
21    return 0;
22
23 }
24
```

The terminal window shows the execution of the program:

```
kali@kali:~/Desktop
$ gcc -g media.c -o media
$ ./media
Inserisci il primo valore intero: 15
Inserisci il secondo valore intero: 7
La media aritmetica e': 11.00
$
```

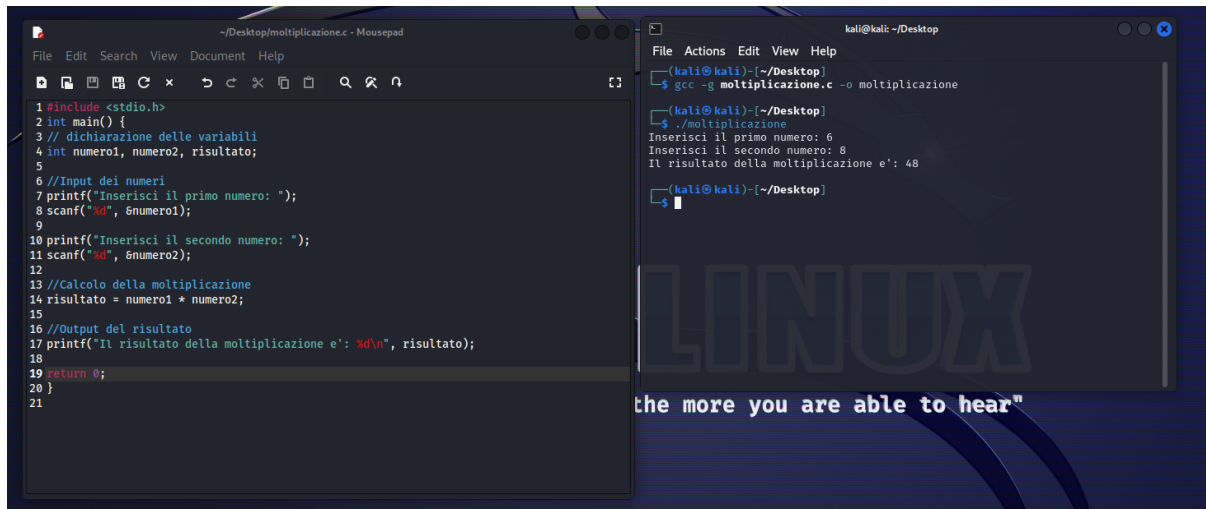
La struttura del linguaggio C:

La prima riga di un programma in linguaggio C riporta sempre la direttiva del processore preceduta dal cancelletto (#), indica inoltre che prima di iniziare a leggere le istruzioni deve caricare in memoria la libreria con funzioni standard del programma C. Nel nostro caso abbiamo usato <stdio.h> che riguarda le funzioni per input e output.

Nella riga 2 riportiamo la variabile int, una variabile che definisce l'insieme di numeri interi corrispondenti a un valore massimo di 4 byte abbinato a Main() che corrisponde alla funzione principale di un programma in C, da qui il processore inizierà a leggere il programma. Inserire il doppio slash (//) ci permette di inserire commenti che la macchina non andrà a leggere. Il blocco di codice è poi definito con le parentesi graffe {}. Bisogna inoltre notare che ogni riga di codice termina con un ";" e dimenticarsi di inserirlo determinerà un errore di sintassi che blocca la lettura del programma. Per inserire stringhe di testo usiamo il comando printf("") e useremo il comando scanf("") per permettere all'utente di inserire i valori nella macchina con cui sta interagendo.

I comandi "%d\n" e "%.2f\n" sono i comandi che andiamo a inserire per permettere alla macchina di eseguire una procedura di calcolo. Nel caso in cui si tratterà di valori numerici

reali INT verrà sostituito con FLOAT (variabile che definisce numeri reali fino a un massimo di 4 byte) e il comando di calcolo sarà “%.fn”. Usando questi comandi saremo in grado di creare programmi di calcolo eseguibili e dove un utente può inserire i valori desiderati.



The image shows a Linux desktop environment with two windows. The left window is a text editor titled "/Desktop/moltiplicazione.c - Mousepad" and contains the following C code:

```
1 #include <stdio.h>
2 int main() {
3 // dichiarazione delle variabili
4 int numero1, numero2, risultato;
5
6 //Input dei numeri
7 printf("Inserisci il primo numero: ");
8 scanf("%d", &numero1);
9
10 printf("Inserisci il secondo numero: ");
11 scanf("%d", &numero2);
12
13 //Calcolo della moltiplicazione
14 risultato = numero1 * numero2;
15
16 //Output del risultato
17 printf("Il risultato della moltiplicazione e': %d\n", risultato);
18
19 return 0;
20 }
21
```

The right window is a terminal titled "kali@kali: ~/Desktop" and shows the compilation and execution of the program:

```
kali@kali:~/Desktop$ gcc -g moltiplicazione.c -o moltiplicazione
kali@kali:~/Desktop$ ./moltiplicazione
Inserisci il primo numero: 6
Inserisci il secondo numero: 8
Il risultato della moltiplicazione e': 48
kali@kali:~/Desktop$
```

In the background, there is a large "LINUX" watermark and the quote "the more you are able to hear" at the bottom right.