

# ESERCITAZIONE PRATICA S2\_L2

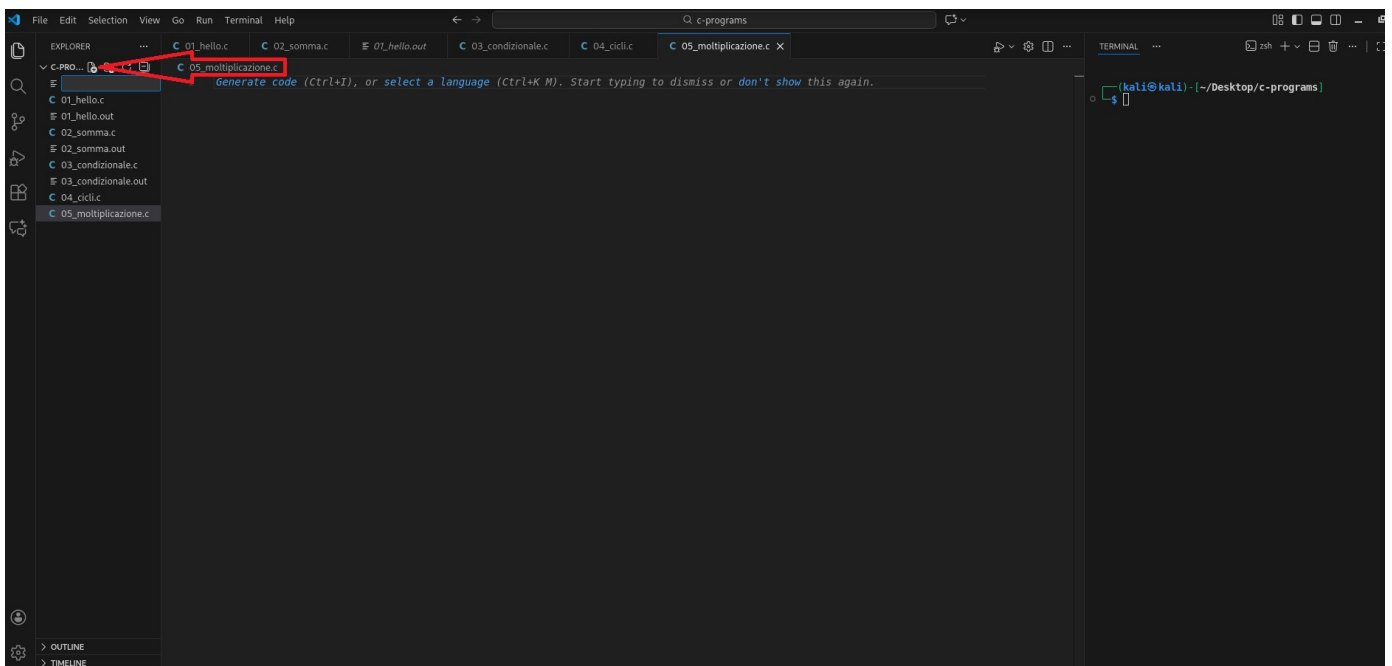
Lo scopo di oggi sarà realizzare due programmi in C:

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

## - Creazione file.c -

Nella nostra schermata di Visual Studio Code all'interno del nostro sistema Linux andiamo a creare un file.c che sarà il nostro script dove andremo a programmare in linguaggio C.

Per creare il nostro *file.c* basterà cliccare nel pulsante in alto a sinistra della sotto-finestra **Explorer** → **New file** → **Rinominare a nostro piacimento**:



Abbiamo generato il nostro script, adesso possiamo iniziare a scrivere codice.

## - Moltiplicazione -

Il nome che ho assegnato allo script è 05\_moltiplicazione.c

```
C 05_moltiplicazione.c > main()
1  #include <stdio.h>
2
3  int main() {
4      int x;
5      int y;
6      int moltiplicazione;
7
8      printf("Inserisci il primo numero: ");
9      scanf("%d" , &x);
10
11     printf("Inserisci il secondo numero: ");
12     scanf("%d" , &y);
13
14     moltiplicazione = x * y;
15
16     printf("La moltiplicazione è: %d" , moltiplicazione);
17
18     return 0;
19 }
```

Per prima cosa da riga 4 a riga 6 vado a definire le mie **variabili** che in questo caso sono tre numeri interi, esse si chiamano: **x**, **y** e **moltiplicazione**.

Utilizzo poi due comandi:

- **printf("esempio");** per poter far printare al prompt ciò che scrivo dentro la **stringa**; (Riga 8, 11 e 16)
- **scanf("%d" , &x);** e **scanf("%d" , &y);** per creare uno "spazio" dove poi andare a scrivere il numero che andremo ad immettere nel prompt quando eseguiremo il programma.

Infine definisco la variabile moltiplicazione (Vedi riga 14).

Il nostro programma è stato creato ma la nostra macchina non è assolutamente in grado di leggere il nostro script.

Per questo motivo andremo a far generare un file .out che non è altro che una versione del nostro codice “tradotta” affinché la macchina capisca i nostri comandi.

Basterà digitare nel prompt questo comando: **gcc -o gcc\_nomefile.out gcc\_nomefile.c**

Si creerà a questo punto un file .out che noi potremo far eseguire alla nostra macchina dal prompt con questo comando: **./nomefile.out**

```
(kali㉿kali)-[~/Desktop/c-programs]
● $ gcc -o 05_moltiplicazione.out 05_moltiplicazione.c

(kali㉿kali)-[~/Desktop/c-programs]
● $ ./05_moltiplicazione.out
Inserisci il primo numero: 5
Inserisci il secondo numero: 2
La moltiplicazione è: 10

(kali㉿kali)-[~/Desktop/c-programs]
○ $
```

## - Media Aritmetica -

La procedura di creazione del **file.c** e **file.out** sono sempre le stesse, ciò che cambia sarà naturalmente il nostro script.

```
C 06_media.c > main()
1  #include <stdio.h>
2
3  int main() {
4      int x;
5      int y;
6      float media;
7
8      printf("Inserisci il primo numero: ");
9      scanf("%d" , &x);
10
11     printf("Inserisci il secondo numero: ");
12     scanf("%d" , &y);
13
14     media = (x + y)/2;
15
16     printf("La media è: %f" , media);
17
18     return 0;
19 }
```

In questo script ho definito gli **int x** e **y** e il **float media**.

Dal momento che il risultato di una media tra due numeri interi potrebbe dare un numero con la virgola allora quella variabile sarà un **float**.

L'operazione vera e propria viene fatta a riga 14 dove calcolo la media tra x e y.

Vediamo adesso dal prompt se il nostro programma funziona:

```
(kali㉿kali)-[~/Desktop/c-programs]
$ gcc -o 06_media.out 06_media.c

(kali㉿kali)-[~/Desktop/c-programs]
$ ./06_media.out
Inserisci il primo numero: 30
Inserisci il secondo numero: 40
La media è: 35.000000

(kali㉿kali)-[~/Desktop/c-programs]
$
```

**Il nostro programma ci restituisce la media esatta!**