

Nel laboratorio di oggi è stato chiesto di:

- scrivere un programma che esegua l'operazione di moltiplicazione tra due numeri inseriti dall'utente.

Di seguito sono riportati gli step principali per la realizzazione del progetto.

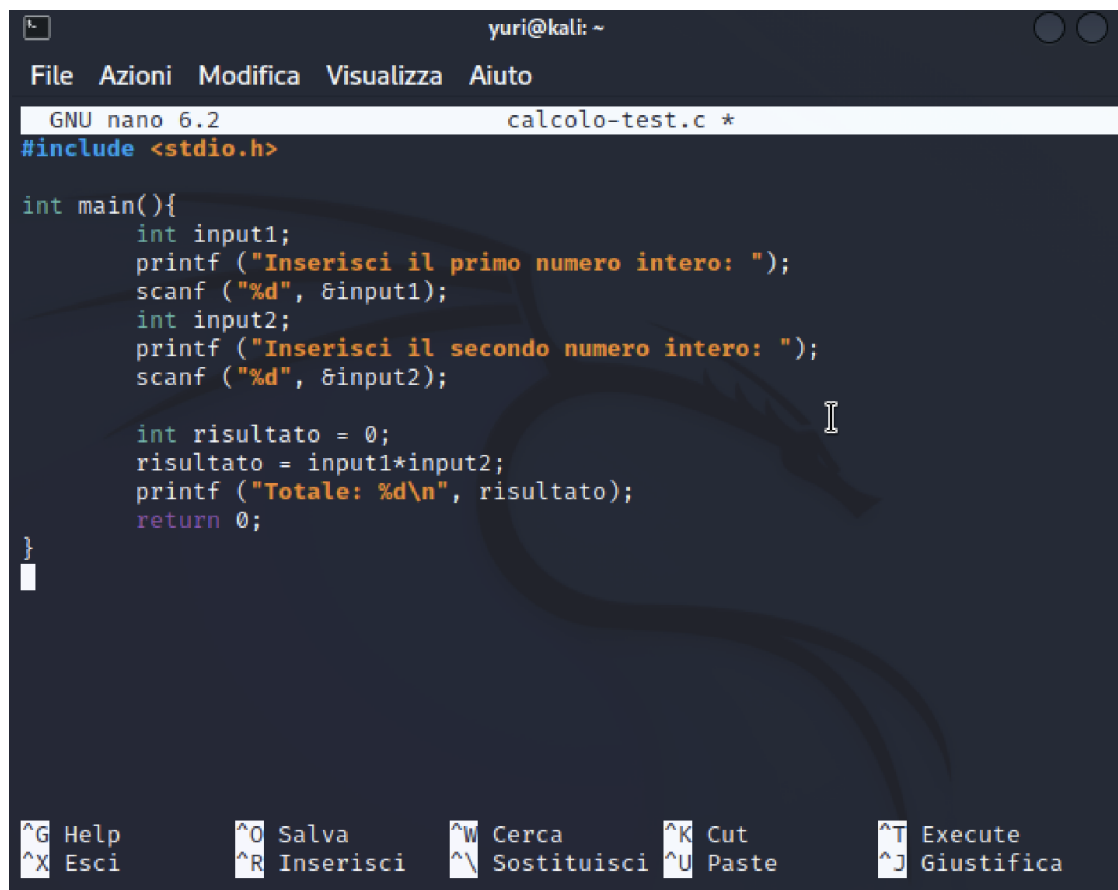
Per iniziare, ho aperto il terminale e scritto i comandi mostrati nell'immagine a destra. Il comando utilizzato serve per creare un file sorgente con estensione .c

Per aprire e modificare il file sorgente appena creato, ho utilizzato il seguente comando:

Una volta premuto **Invio**, si aprirà una finestra con l'editor per iniziare a scrivere o modificare il codice.

```
(yuri@kali)-[~]  
$ touch calcolo-test.c
```

```
(yuri@kali)-[~]  
$ nano calcolo-test.c
```



```
yuri@kali: ~  
File Azioni Modifica Visualizza Aiuto  
GNU nano 6.2          calcolo-test.c *  
#include <stdio.h>  
  
int main(){  
    int input1;  
    printf ("Inserisci il primo numero intero: ");  
    scanf ("%d", &input1);  
    int input2;  
    printf ("Inserisci il secondo numero intero: ");  
    scanf ("%d", &input2);  
  
    int risultato = 0;  
    risultato = input1*input2;  
    printf ("Totale: %d\n", risultato);  
    return 0;  
}
```

Help Esci Salva Inserisci Cerca Sostituisci Cut Paste Execute Giustifica

Nell'immagine precedente, si possono notare alcuni elementi essenziali per il corretto funzionamento del programma in C:

1. **Inclusione delle librerie necessarie:** L'istruzione `#include <stdio.h>` è obbligatoria per permettere l'uso delle funzioni di input e output, come `printf` e `scanf`.
2. **Definizione della funzione main:** La funzione `int main()` è la funzione principale dove inizia l'esecuzione del programma.

### 3. Dichiarazione delle variabili:

- `int input1, input2`: due variabili intere che l'utente utilizza per inserire i numeri da moltiplicare.
- `int risultato = 0`: una variabile per memorizzare il risultato della moltiplicazione, inizializzata a 0.

### 4. Input dell'utente:

- Si utilizza `scanf` per leggere i due numeri inseriti dall'utente e assegnarli a `input1` e `input2`.

### 5. Calcolo del risultato:

- `risultato = input1 * input2`: il primo numero viene moltiplicato per il secondo e il risultato viene salvato nella variabile `risultato`.

### 6. Output del risultato:

- Si utilizza `printf` per stampare il risultato della moltiplicazione.

### 7. Termine del programma:

- La funzione `return 0`; indica la fine dell'esecuzione del programma.

### Salvataggio del file

Una volta terminato il codice, è necessario salvare il file, nel mio caso, premendo **Command + S**. Dopo aver salvato, si può procedere con il passaggio successivo, che consiste nella compilazione ed esecuzione del programma.

Dopo aver scritto e salvato il codice nel file è necessario compilare il programma utilizzando il comando `gcc`. Per farlo, si deve eseguire il comando:

```
(yuri@kali)-[~]  
$ gcc calcolo-test.c -o calcolo-test
```

### Ultimo passaggio: esecuzione del programma e verifica del risultato

Per avviare il programma e testare la sua funzionalità, si deve utilizzare il comando:

```
(yuri@kali)-[~]  
$ ./calcolo-test  
Inserisci il primo numero intero: 30  
Inserisci il secondo numero intero: 89  
Totale: 2670  
  
(yuri@kali)-[~]  
$
```

Dopo aver eseguito il comando, il programma richiederà di inserire i numeri da moltiplicare. Inseriamo i numeri, ad esempio 30 e 80, e il programma calcolerà automaticamente il risultato.

