

martedì 3 dicembre 2024

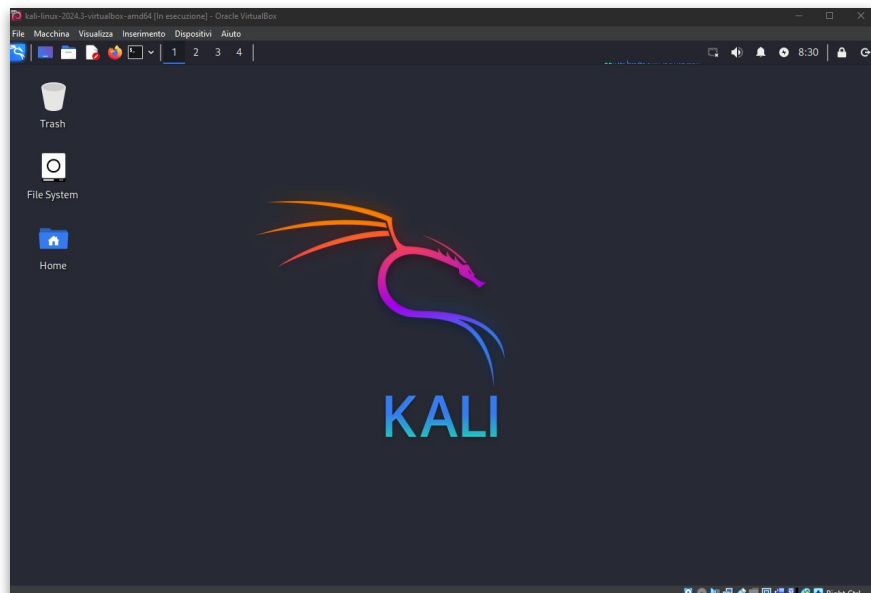
S2/L2 - Programmazione in C

L'esercizio di oggi prevede la realizzazione di 2 programmi in **C**:

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

ESERCIZIO 1

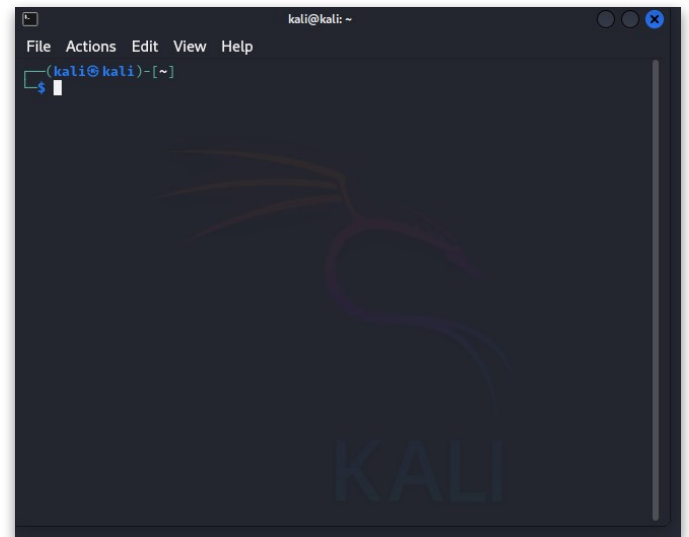
- Innanzitutto apro la VM di Kali Linux in VirtualBox e la avvio, poi faccio l'accesso e mi trovo nel desktop:



PROCEDIMENTO

1. APRO IL TERMINALE

- Con [Ctrl+Alt+T] apro il **terminale**:

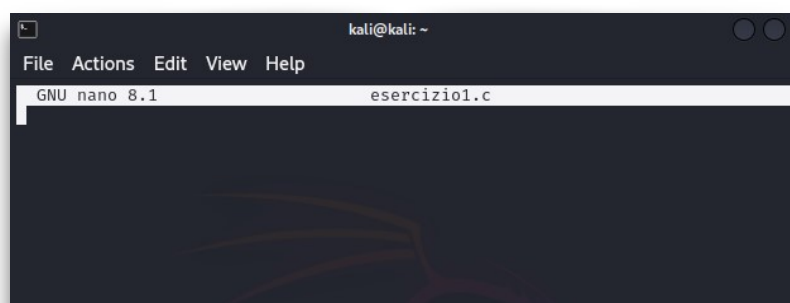


2. CREO IL FILE CON ESTENSIONE .C



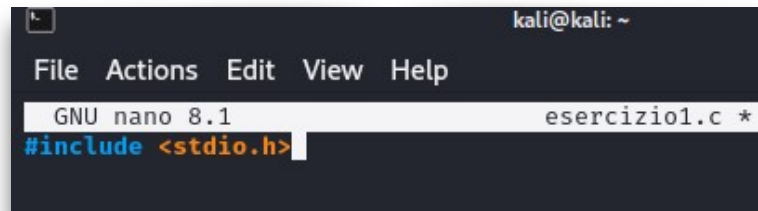
- Per creare il file scrivo:
“**nano esercizio1.c**” e invio:

- Si apre l'**editor** di testo dove scrivo il codice:



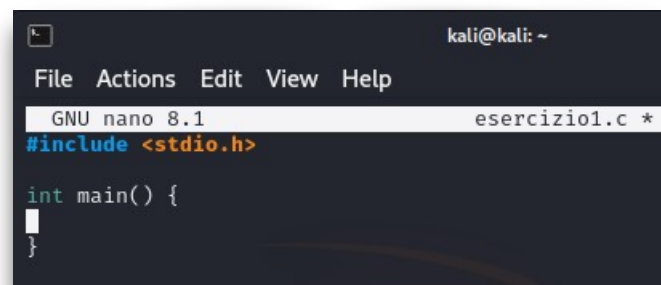
3. SCRITTURA DEL PROGRAMMA

- Scrivo “**#include <stdio.h>**”, che mi serve per includere la libreria standard per input e output:



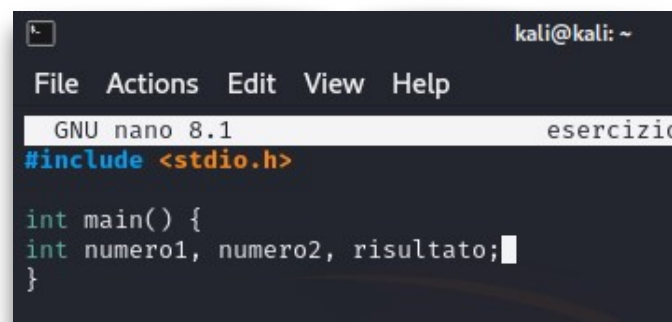
A screenshot of the nano text editor running on a Kali Linux system. The title bar shows 'kali@kali: ~'. The menu bar includes 'File', 'Actions', 'Edit', 'View', and 'Help'. The status bar at the bottom indicates 'GNU nano 8.1' and 'esercizio1.c *'. The main editing area contains the text `#include <stdio.h>` with a cursor at the end of the line.

- Scrivo “**int main()**” che segna il punto di inizio del programma:



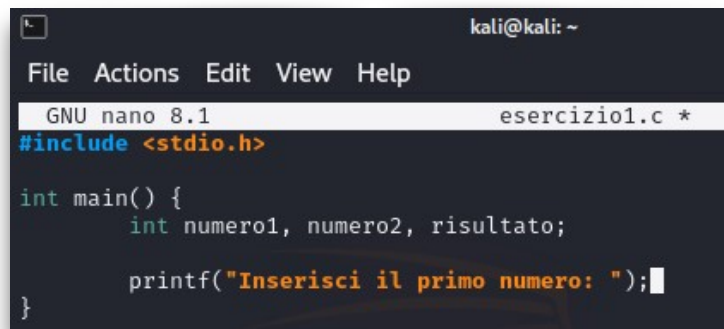
A screenshot of the nano text editor. The status bar shows 'esercizio1.c *'. The code now includes `#include <stdio.h>` followed by a new line containing `int main() {`. The cursor is positioned at the start of the next line, which is currently empty.

- Ora includo le variabili digitando “**int numero1, numero2, risultato**”, dove numero1 e numero2 vengono inseriti dall’utente e risultato è banalmente il risultato della moltiplicazione dei due numeri:



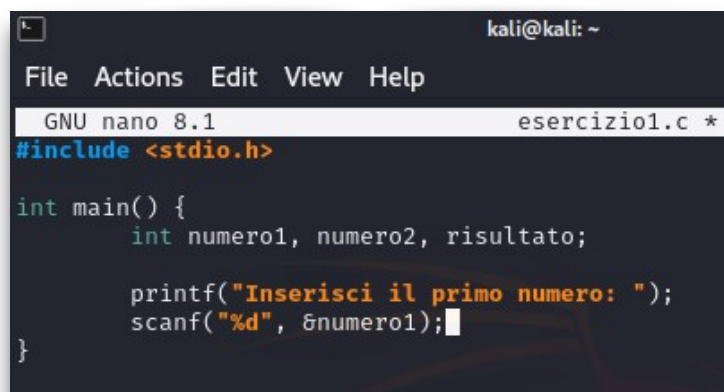
A screenshot of the nano text editor. The code now includes `#include <stdio.h>`, `int main() {`, and a new line with `int numero1, numero2, risultato;`. The cursor is at the end of this line.

- Uso “**printf()**” per indicare all’utente di inserire il primo numero:



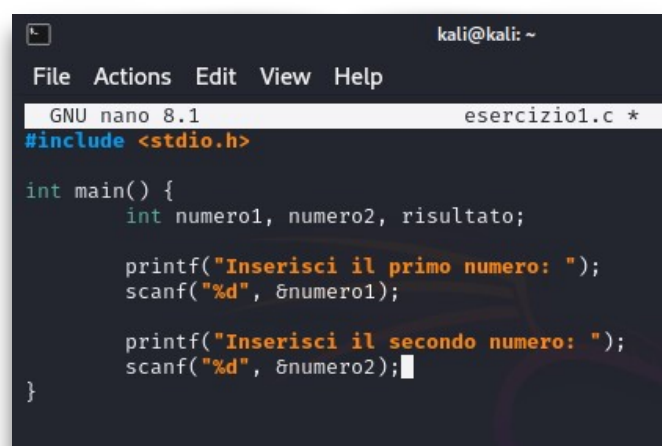
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.1 esercizio1.c *  
#include <stdio.h>  
  
int main() {  
    int numero1, numero2, risultato;  
  
    printf("Inserisci il primo numero: ");  
}
```

- Uso “**scanf()**” per leggere l’input dell’utente e con “%d” indico che il formato deve essere un intero, “&” è necessario perché il valore venga memorizzato come numero1:




```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.1 esercizio1.c *  
#include <stdio.h>  
  
int main() {  
    int numero1, numero2, risultato;  
  
    printf("Inserisci il primo numero: ");  
    scanf("%d", &numero1);  
}
```

- Ora faccio lo stesso per il secondo numero:



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.1 esercizio1.c *  
#include <stdio.h>  
  
int main() {  
    int numero1, numero2, risultato;  
  
    printf("Inserisci il primo numero: ");  
    scanf("%d", &numero1);  
  
    printf("Inserisci il secondo numero: ");  
    scanf("%d", &numero2);  
}
```

- A questo punto indico al programma che la variabile risultato è data dalla moltiplicazione di numero1 per numero2:



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 esercizio1.c *
#include <stdio.h>

int main() {
    int numero1, numero2, risultato;

    printf("Inserisci il primo numero: ");
    scanf("%d", &numero1);

    printf("Inserisci il secondo numero: ");
    scanf("%d", &numero2);

    risultato = numero1 * numero2;
}
```

- Con “**printf()**” stampo il risultato della moltiplicazione, uso “**%d**” per stampare un numero intero:



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 esercizio1.c *
#include <stdio.h>

int main() {
    int numero1, numero2, risultato;

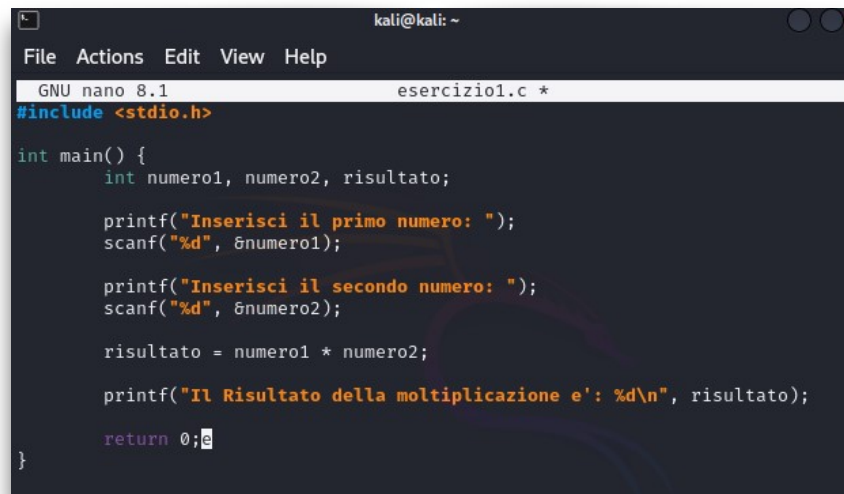
    printf("Inserisci il primo numero: ");
    scanf("%d", &numero1);

    printf("Inserisci il secondo numero: ");
    scanf("%d", &numero2);

    risultato = numero1 * numero2;

    printf("Il Risultato della moltiplicazione e': %d\n", risultato);
}
```

- Infine scrivo “**return 0**” e il programma è completo:



```
GNU nano 8.1 esercizio1.c *
#include <stdio.h>

int main() {
    int numero1, numero2, risultato;

    printf("Inserisci il primo numero: ");
    scanf("%d", &numero1);

    printf("Inserisci il secondo numero: ");
    scanf("%d", &numero2);

    risultato = numero1 * numero2;

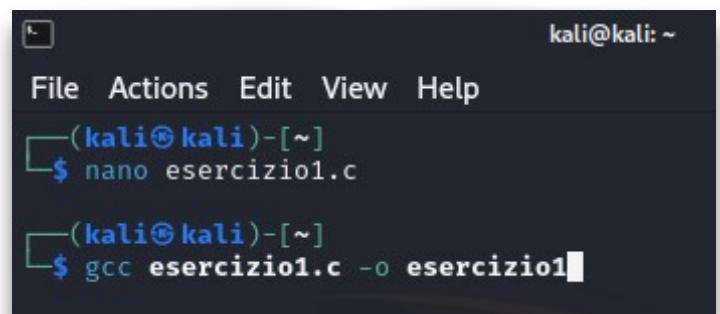
    printf("Il Risultato della moltiplicazione e': %d\n", risultato);

    return 0;
}
```

Salvo il file con [Ctrl+O] e invio, poi con [Ctrl+X] esco dall'editor.

4. COMPILAZIONE DEL PROGRAMMA

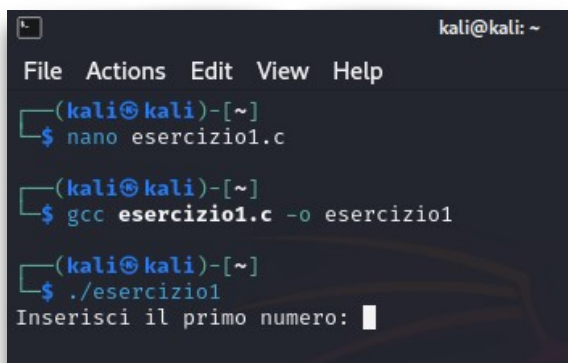
- Per compilarlo lancio il comando “**gcc esercizio1.c -o esercizio1**”:



```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nano esercizio1.c

(kali@kali)-[~]
$ gcc esercizio1.c -o esercizio1
```

5. ESECUZIONE DEL PROGRAMMA



```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nano esercizio1.c

(kali@kali)-[~]
$ gcc esercizio1.c -o esercizio1

(kali@kali)-[~]
$ ./esercizio1
Inserisci il primo numero: 
```

- Per eseguirlo scrivo “**./esercizio1**”:

- Ora inserisco i numeri, ad esempio 5 e 3:

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nano esercizio1.c  
(kali@kali)-[~]  
$ gcc esercizio1.c -o esercizio1  
(kali@kali)-[~]  
$ ./esercizio1  
Inserisci il primo numero: 5  
Inserisci il secondo numero: 3  
Il Risultato della moltiplicazione e': 15
```

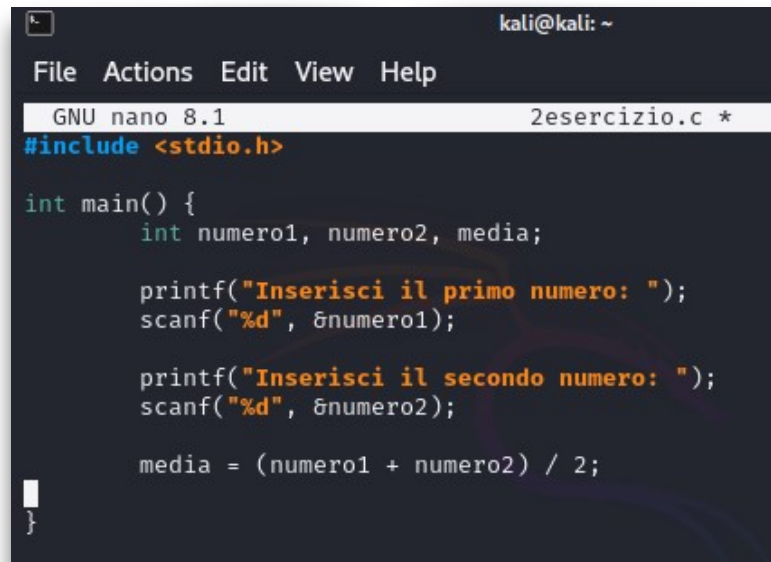
Come si vede da come risultato 15 che è corretto, così abbiamo dimostrato che funziona!

ESERCIZIO 2

- Per il secondo esercizio seguo gli stessi passi fino alle istruzioni per il calcolo, a parte per il nome del progetto che chiamo “**2esercizio**” e per la variabile risultato che chiamo “**media**”.
- Quindi ci troviamo a questo punto:

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.1 2esercizio.c *  
#include <stdio.h>  
  
int main() {  
    int numero1, numero2, media;  
  
    printf("Inserisci il primo numero: ");  
    scanf("%d", &numero1);  
  
    printf("Inserisci il secondo numero: ");  
    scanf("%d", &numero2);
```

- Ora dico al programma il calcolo da eseguire per trovare la media ovvero “**media = (numero1 + numero2) / 2**”:



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 2esercizio.c *
#include <stdio.h>

int main() {
    int numero1, numero2, media;

    printf("Inserisci il primo numero: ");
    scanf("%d", &numero1);

    printf("Inserisci il secondo numero: ");
    scanf("%d", &numero2);

    media = (numero1 + numero2) / 2;

}
```

- A questo punto la parte finale è come quella dell'esercizio precedente:



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 2esercizio.c *
#include <stdio.h>

int main() {
    int numero1, numero2, media;

    printf("Inserisci il primo numero: ");
    scanf("%d", &numero1);

    printf("Inserisci il secondo numero: ");
    scanf("%d", &numero2);

    media = (numero1 + numero2) / 2;

    printf("La media dei due numeri e': %d\n", media)

    return 0;
}
```


- Compilo e poi eseguo:

!Sembra esserci un errore quindi torno al codice e lo controllo, e mi accorgo che mancava un “;” a chiudere la stringa:

```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.1 2esercizio.c *
#include <stdio.h>

int main() {
    int numero1, numero2, media;

    printf("Inserisci il primo numero: ");
    scanf("%d", &numero1);

    printf("Inserisci il secondo numero: ");
    scanf("%d", &numero2);

    media = (numero1 + numero2) / 2;

    printf("La media dei due numeri e': %d\n", media);
    return 0;
}
```

Ora riprovo a compilarlo ed eseguo:

```
kali@kali: ~
File Actions Edit View Help
$ nano 2esercizio.c
(kali@kali)-[~]
$ gcc 2esercizio.c -o 2esercizio
2esercizio.c: In function 'main':
2esercizio.c:14:58: error: expected ';' before 'return'
14 |         printf("La media dei due numeri e': %d\n", media)
    |                                                         ^
15 |
16 |         return 0;
    |         ~~~~~
(kali@kali)-[~]
$ nano 2esercizio.c
(kali@kali)-[~]
$ gcc 2esercizio.c -o 2esercizio
(kali@kali)-[~]
$ ./2esercizio
Inserisci il primo numero: 10
Inserisci il secondo numero: 5
La media dei due numeri e': 7
```

Il programma funziona correttamente!