

Traccia:

1 – Si scriva un programma che esegua l' operazione moltiplicazione tra due numeri inseriti dall' utente.

Si richiede di scrivere un programma in linguaggio C utilizzando Kali Linux.

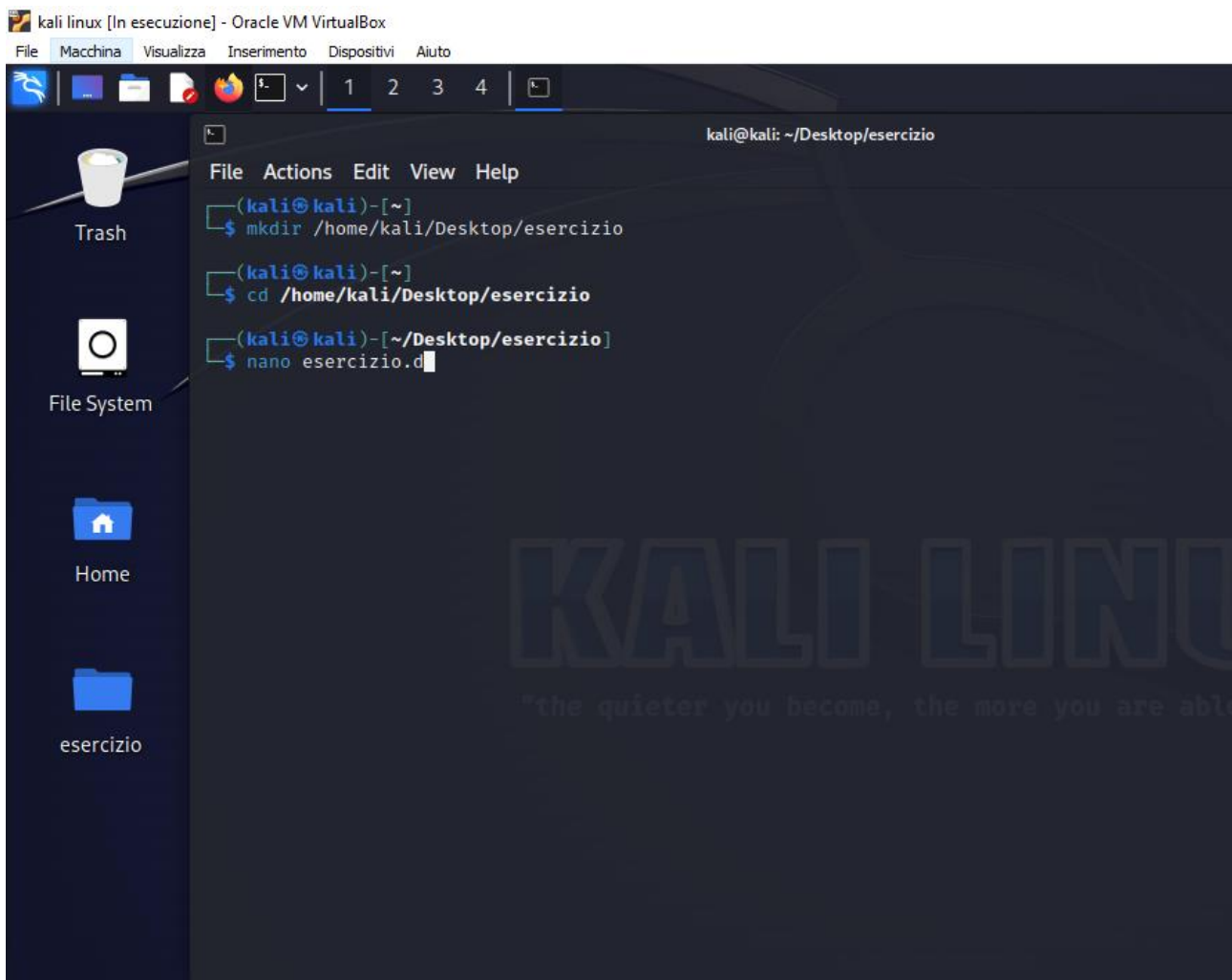
Procedo quindi su terminale a creare una cartella esempio e dentro un file di testo con cui programmare.

Per creare la cartella uso il comando “ **mkdir /home/kali/Desktop/esercizio**

Poi mi sposto sulla cartella appena creata “esempio” attraverso il comando “**cd /home/kali/Desktop/esercizio .**

Poi procedo a con il comando “**nano esercizio.c**” in modo da aprire il file creato e poterlo editare, cioè usarlo per scrivere il programma.

E' importante usare l'estensione .c del file in modo che il compilatore (gcc) possa riconoscere il file.



The screenshot shows a Kali Linux virtual machine running on Oracle VM VirtualBox. The window title is "kali linux [In esecuzione] - Oracle VM VirtualBox". The menu bar includes "File", "Macchina", "Visualizza", "Inserimento", "Dispositivi", and "Aiuto". The toolbar shows icons for file operations and navigation. The main window is split into two panes. The left pane, titled "~/Desktop/esercizio/esercizio.c - Mousepad", contains a C program. The right pane, titled "kali@kali: ~/Desktop/esercizio", shows the terminal output of the program's execution.

```
1 #include <stdio.h>
2
3 int main()
4 {
5 {
6
7 int primo_numero;
8 int secondo_numero;
9 int moltiplicazione;
10
11 printf ("inserisci il primo numero:\n");
12 scanf ("%d", &primo_numero);
13 printf ("inserisci il secondo numero:\n");
14 scanf ("%d", &secondo_numero);
15 moltiplicazione= primo_numero*secondo_numero;
16
17 printf ("la moltiplicazione dei due numeri inseriti è:%d\n",
    moltiplicazione);
18
19
20 return 0;
21
22 }
```

The terminal output on the right shows the following sequence of commands and results:

```
(kali@kali)-[~/Desktop/esercizio]
$ gcc -g esercizio.c -o esercizio
(kali@kali)-[~/Desktop/esercizio]
$ ./esercizio
inserisci il primo numero:
10
inserisci il secondo numero:
5
la moltiplicazione dei due numeri inseriti è:50
(kali@kali)-[~/Desktop/esercizio]
$
```

Una volta nel file ho scritto il programma. Intanto ho impostato di includere la **libreria stdio.h** la quale fornisce le funzioni di input e output standard, e della quale andiamo ad utilizzare le funzioni printf e scanf.

Poi ho inserito la funzione principale che troviamo in tutti i linguaggi di programmazione, ovvero **"int main ()"**, che indica la funzione principale (main), il suo tipo (int=intero) e () che indica che main è una funzione, senza non si tratterebbe di una funzione.

Poi inserisco le **parentesi graffe** che racchiudono il blocco del codice.

Inserisco poi dichiaro al calcolatore le variabili da ricordare e poi utilizzare nella moltiplicazione, che sono numeri interi (int).

Al rigo 11 la funzione printf (che devi fare) stampa a video, dice a chi utilizza il programma la prima azione da fare, in questo caso specifico assegnare un valore, cioè un numero, alla **prima variabile (primo_numero)**, che ho inserito successivamente (10).

Poi al rigo 12 Scanf (si aspetta ciò che è contenuto nella funzione) sospende il programma in attesa di un input da parte di chi utilizza il programma. Infatti con la formula : **scanf ("%d", &primo_numero)**, stiamo dicendo a scanf che deve aspettarsi un valore (che è intero %d, (10)), e che quel valore deve essere assegnato alla variabile scritta dopo il puntatore **&**, nel nostro caso primo_numero.

Al rigo 13 e 14 la logica è la stessa ma stavolta con printf comunichiamo all'utente che vogliamo come input un altro numero che poi verrà associato da scanf alla **variabile int secondo_numero**.

Al rigo 15 si dice al calcolatore il valore che deve essere assegnato alla **variabile moltiplicazione** , che nel nostro caso è la moltiplicazione del primo numero con il secondo numero.

Il **rigo 17** si dice al calcolatore di (printf) stampare all'utente il risultato della moltiplicazione del primo e secondo numero, i cui valori l'utente (io in questo caso) ho inserito (rispettivamente 10 e 5)

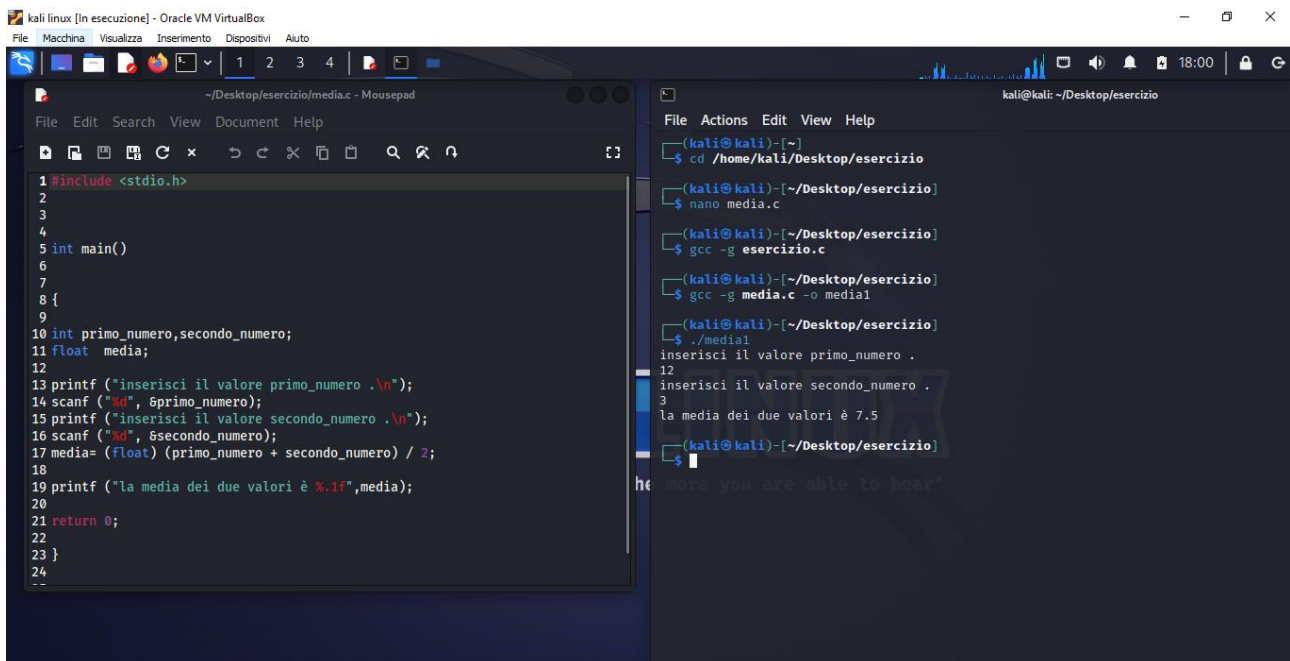
Al rigo 20 con **Return 0** concludo la funzione, perché altrimenti andrebbe avanti all'infinito. In pratica gli stiamo comunicando che la funzione per cui abbiamo scritto il programma si è concluso con la stampa del risultato della moltiplicazione e che può chiudere tutto.

Prima mi sposto sulla cartella "esercizio" che ho creato attraverso il terminale con il comando **cd /home/kali/Desktop/esercizio**.

Con il comando **gcc -g esercizio.c -o esercizio** ho compilato il file esercizio.c e a renderlo eseguibile con il file chiamato esercizio.

Infine con il comando **./esercizio** eseguiamo il file, cioè diamo esecuzione al programma. Come ci si aspetta, il programma chiede di inserire due numeri, e va poi a stampare il loro prodotto.

2 – Si scriva un programma in linguaggio C che legga due valori interi e visualizzi la loro media aritmetica.



```
~/Desktop/esercizio/media.c - Mousepad
File Edit Search View Document Help
1 #include <stdio.h>
2
3
4
5 int main()
6
7
8 {
9
10 int primo_numero,secondo_numero;
11 float media;
12
13 printf ("inserisci il valore primo_numero .\n");
14 scanf ("%d", &primo_numero);
15 printf ("inserisci il valore secondo_numero .\n");
16 scanf ("%d", &secondo_numero);
17 media= (float) (primo_numero + secondo_numero) / 2;
18
19 printf ("la media dei due valori è %.1f",media);
20
21 return 0;
22
23 }
24
--

kali@kali: ~/Desktop/esercizio
File Actions Edit View Help
(kali@kali)-[~]
$ cd /home/kali/Desktop/esercizio
(kali@kali)-[~/Desktop/esercizio]
$ nano media.c
(kali@kali)-[~/Desktop/esercizio]
$ gcc -g esercizio.c
(kali@kali)-[~/Desktop/esercizio]
$ gcc -g media.c -o medial
(kali@kali)-[~/Desktop/esercizio]
$ ./medial
inserisci il valore primo_numero .
12
inserisci il valore secondo_numero .
3
la media dei due valori è 7.5
(kali@kali)-[~/Desktop/esercizio]
$
```

La seconda traccia chiede invece di eseguire un programma che legga valori e faccia la media aritmetica.

Ancora una volta al rigo 1 ho incluso la libreria “**stdio.h**” che è quella che contiene e gestisce le funzione di printf e scanf.

Dopo aver richiamato la funzione principale **main**, specificandone il tipo **int** (intero) ho messo le parentesi graffe, aperte e chiuse, che delineano il blocco del codice.

Ho poi dichiarato le variabili intere primo e secondo numero e la terza variabile reale (float=è il dato che indica i valori reali) stavolta è rappresentata dalla media aritmetica.

Al rigo 13 la funzione printf (generalmente indica al programma di stampare quanto inserisco nelle virgolette) dice a chi utilizza il programma cosa deve fare, cioè assegnare un valore alla **prima variabile** primo_numero. Con il rigo successivo scanf sospende il programma in attesa dell’input dell’utente e con la funzione comunichiamo che deve aspettarsi un valore intero (%d), e che dovrà associarlo (questo si vede tramite il puntatore &) alla variabile “primo_numero”.

Alle righe 15 e 16 la logica è la stessa ma stavolta la **variabile** coinvolta è la “secondo_numero”.

Al rigo 17 si comunica al calcolatore il valore da attribuire alla **variabile media** che è la somma aritmetica delle due variabili diviso due (andiamo ad associare il numero medio delle due variabili, quindi sommiamo e poi le dividiamo per due). Il float dice di utilizzare i due numeri successivi come se fossero reali e non interi e a quel punto andando a considerare la somma nei reali e i valori nei reali vengono considerati i valori dopo la virgola.

Al rigo 19 comunichiamo alla funzione printf di stampare il risultato della media aritmetica dei due valori che l’utente (io in questo caso) associa alle variabili primo_numero e secondo_numero, e che questo

risultato deve (con il puntatore **%**) essere un numero reale (**f**) (hanno la virgola), specificando di inserire un solo numero dopo la virgola (**1**)

Poi con **return 0** concludo il programma e con la parentesi graffa chiudo il blocco del codice (**}**).

Ho salvato poi il file con **ctrl + x**, ho digitato **y** e poi **avvio**.

Prima mi sposto sulla cartella “esercizio” che ho creato attraverso il terminale (nuova pagina aperta) con il comando **cd /home/kali/Desktop/esercizio**, e dopo sul file **media.c** con il comando **nano**.

Con il comando **gcc -g media.c -o media** ho compilato il file **media.c** e l’ho reso eseguibile con il file chiamato **media**, quindi assegnando un altro nome.

Infine con il comando **./media1** eseguiamo il file, cioè diamo esecuzione al programma. Come ci si aspetta, il programma chiede di inserire due numeri, e va poi a stampare la loro media (7,5) che è appunto un numero reale.