

Questo lavoro io l'ho svolto con Kali Linux, quindi partiamo dal scrivere il codice con lo strumento "mousepad text editor" ed andiamo a spiegare il codice.

Esercizio 1

```
1 #include <stdio.h>
2
3 int main() {
4     // Dichiarazione delle variabili intere
5     int numero1, numero2, risultato;
6
7     // Richiesta all'utente di inserire il primo numero
8     printf("Inserisci il primo numero: ");
9     scanf("%d", &numero1);
10
11    // Richiesta all'utente di inserire il secondo numero
12    printf("Inserisci il secondo numero: ");
13    scanf("%d", &numero2);
14
15    // Calcolo della moltiplicazione
16    risultato = numero1 * numero2;
17
18    // Stampare il risultato
19    printf("Il risultato della moltiplicazione e': %d\n", risultato);
20
21    return 0;
22 }
```

La prima riga di codice `#include <stdio.h>` serve a capire che libreria stiamo utilizzando ed a far capire alla macchina che linguaggio stiamo usando, in questo caso C. Senza non capirà nulla del codice. Utilizzerò spesso la funzione di commento `//` che verrà automaticamente ignorata dalla macchina e serve solo a me per capire meglio la situazione.

Dichiarazione di tre variabili intere:

numero1, numero2 e risultato. Queste variabili sono utilizzate per memorizzare i numeri inseriti dall'utente e il risultato della moltiplicazione. Ricordiamo che ogni volta che finisco una riga devo mettere ;

Richiesta all'utente di inserire il primo numero e secondo numero:

Stampa un messaggio per chiedere all'utente di inserire il primo numero intero con **printf** e quindi utilizza **scanf** per leggere l'input e memorizzarlo nella variabile numero1 e faccio lo stesso passaggio col numero2. Il comando **%d** è una specifica per stampare il messaggio in numero intero. Il comando **&** viene utilizzato per memorizzare il valore inserito dall'utente nella variabile messa dopo il segno.

Calcolo della moltiplicazione: Semplicemente vado a dare al risultato la moltiplicazione tra i due numeri.

Return 0: Indica che il programma è stato eseguito correttamente. Il valore 0 restituito al sistema operativo indica che il programma è terminato correttamente. Se fosse 1 è successo un errore.
(scendi sotto per l'esercizio 2)

Esercizio 2

Andiamo a vedere nel secondo le cose che non ho spiegato dato che questo è un po' diverso.

```
1 #include <stdio.h>
2
3 int main() {
4     // Dichiarazione delle variabili intere
5     int numero1, numero2;
6     float media;
7
8     // Richiesta all'utente di inserire il primo numero
9     printf("Inserisci il primo numero intero: ");
10    scanf("%d", &numero1);
11
12    // Richiesta all'utente di inserire il secondo numero
13    printf("Inserisci il secondo numero intero: ");
14    scanf("%d", &numero2);
15
16    // Calcolo della media aritmetica
17    media = (float)(numero1 + numero2) / 2;
18
19    // Stampare la media (perchè è così difficile...)
20    if (media == (int)media) {
21        printf("La media aritmetica di e' %d\n", (int)media);
22    } else {
23        printf("La media aritmetica di e' %.1f\n", media);
24    }
25
26    return 0;
27 }
28 |
```

Dichiarazione di tre variabili: numero1 e numero2 sono interi, media è un float. Float significa numeri reali, pertanto lo uso per avere anche i numeri decimali.

Calcola la media aritmetica: la somma di numero1 e numero2 viene prima convertita in un float (float) per assicurarsi che la divisione successiva generi un risultato con la parte decimale.

Stampare la media (complicatuccio): Questa sezione del codice utilizza un'istruzione condizionale (if-else) per determinare come stampare il risultato della media.

if (media == (int)media): Questa condizione verifica se la variabile `media` è uguale alla sua rappresentazione come intero. Se la parte decimale di `media` è zero, significa che la `media` è un numero intero. In questo caso, l'istruzione dentro l'`if` viene eseguita.

Qui viene utilizzato il formato `%d` per stampare il risultato come un numero intero senza la parte decimale. `(int)media` viene utilizzato per ottenere la parte intera di `media` e stamparla come un intero.

else: Se la condizione nell'`if` non è soddisfatta (quindi la `media` ha una parte decimale diversa da zero), l'istruzione nell'`else` viene eseguita.

else: Se la condizione nell'`if` non è soddisfatta (quindi la `media` ha una parte decimale diversa da zero), l'istruzione nell'`else` viene eseguita. Qui, viene utilizzato il formato `%f` per stampare il risultato con numeri reali, io ho utilizzato `%.1f` così mi mostra solo una cifra dopo la virgola, se è `%.2f` mi mostra due cifre dopo la virgola e così via.