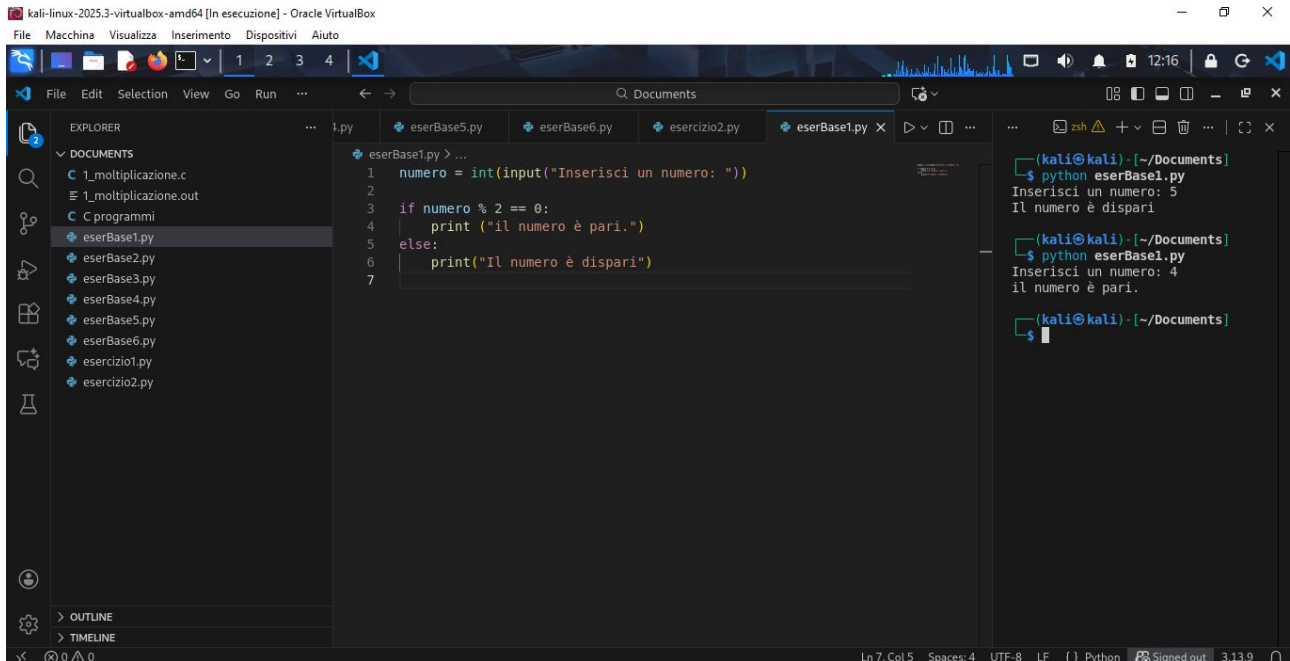


## ESERCITAZIONE DEL 3.12

### ESERCIZIO 1 NUMERO PARI O DISPARI



The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The file explorer on the left shows a directory named 'DOCUMENTS' containing several Python files, including 'eserBase1.py'. The main editor window displays the code for 'eserBase1.py', which prompts the user to enter a number and checks if it is even or odd using a modulo operation. The terminal on the right shows the execution of the script twice: first with the input '5' resulting in 'Il numero è dispari', and then with the input '4' resulting in 'Il numero è pari'.

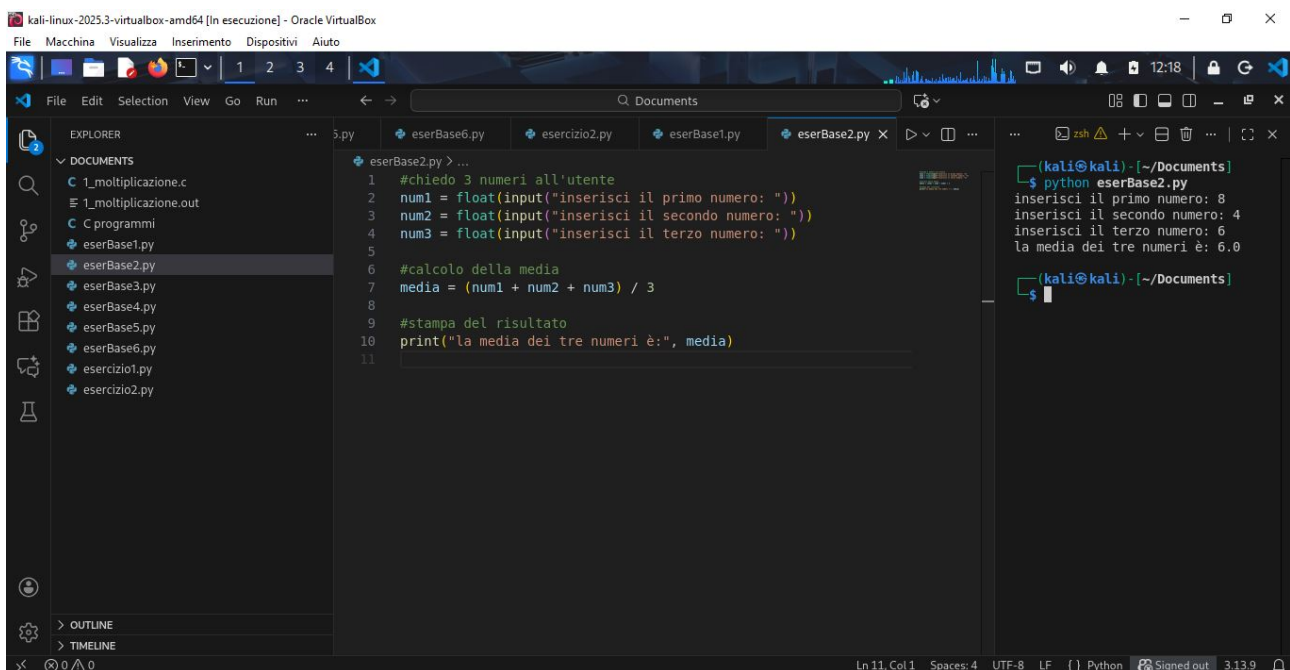
```
1 numero = int(input("Inserisci un numero: "))
2
3 if numero % 2 == 0:
4     print ("il numero è pari.")
5 else:
6     print("Il numero è dispari")
7
```

```
(kali@kali) - [~/Documents]
$ python eserBase1.py
Inserisci un numero: 5
Il numero è dispari

(kali@kali) - [~/Documents]
$ python eserBase1.py
Inserisci un numero: 4
Il numero è pari.

(kali@kali) - [~/Documents]
$
```

### ESERCIZIO 2 CALCOLO DELLA MEDIA



The screenshot shows the same Kali Linux virtual machine with the Visual Studio Code editor. The file explorer now highlights 'eserBase2.py'. The main editor window displays the code for 'eserBase2.py', which prompts the user to enter three numbers and calculates their average. The terminal on the right shows the execution of the script, where the user enters the numbers 8, 4, and 6, resulting in the output 'la media dei tre numeri è: 6.0'.

```
1 #chiedo 3 numeri all'utente
2 num1 = float(input("inserisci il primo numero: "))
3 num2 = float(input("inserisci il secondo numero: "))
4 num3 = float(input("inserisci il terzo numero: "))
5
6 #calcolo della media
7 media = (num1 + num2 + num3) / 3
8
9 #stampa del risultato
10 print("la media dei tre numeri è:", media)
11
```

```
(kali@kali) - [~/Documents]
$ python eserBase2.py
inserisci il primo numero: 8
inserisci il secondo numero: 4
inserisci il terzo numero: 6
la media dei tre numeri è: 6.0

(kali@kali) - [~/Documents]
$
```

### ESERCIZIO 3 TABELLINA DI UN NUMERO

The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The Explorer panel on the left shows a file tree with documents. The main editor window displays a Python script named `eserBase3.py`. The script prompts the user for a number and prints a multiplication table for that number from 1 to 10. The output in the terminal shows the table for the number 8.

```
eserBase3.py > ...
1 #chiedi un numero all'utente
2
3 numero = int(input("Inserisci un numero: "))
4
5 #stampa la tabellina fino a 10
6 for i in range(1, 11):
7     risultato = numero * i
8     print(numero, "x", i, "=", risultato)
```

```
(kali@kali) - [~/Documents]
$ python eserBase3.py
Inserisci un numero: 8
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80

(kali@kali) - [~/Documents]
$
```

## ESERCIZIO 4 CONTARE VOCALI IN UNA STRINGA

The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The Explorer panel on the left shows a file tree with documents. The main editor window displays a Python script named `eserBase4.py`. The script prompts the user for a string and counts the number of vowels (both uppercase and lowercase) in the string. The output in the terminal shows the count for the string "contatore", which is 5.

```
eserBase4.py > ...
1 #chiede una stringa all'utente
2 testo = input("Inserisci una stringa: ")
3
4 #insieme delle vocali (maiuscole e minuscole)
5 vocali = "aeiouAEIOU"
6
7 #contatore iniziale
8 conta = 0
9
10 #scorre ogni carattere della stringa
11 for carattere in testo:
12     if carattere in vocali:
13         conta += 1
14
15 #stampa il risultato
16 print("numero di vocali presenti:", conta)
```

```
(kali@kali) - [~/Documents]
$ python eserBase4.py
Inserisci una stringa: contatore
numero di vocali presenti: 5

(kali@kali) - [~/Documents]
$
```

## ESERCITAZIONE 5 TROVA IL MASSIMO IN UNA LISTA

The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The Explorer panel on the left shows a directory structure with files like `1_moltiplicazione.c`, `eserBase1.py`, `eserBase2.py`, `eserBase3.py`, `eserBase4.py`, `eserBase5.py`, `eserBase6.py`, `esercizio1.py`, and `esercizio2.py`. The main editor window displays the code for `eserBase5.py`:

```
1 #chiedo una lista di numeri separati da spazi
2 numeri_input = input("Inserisci una lista di numeri separati da spazi: ")
3
4 #converto la stringa in lista numeri
5 numeri = [float(n) for n in numeri_input.split()]
6
7 #trova il massimo
8 massimo = max(numeri)
9
10 print("il numero di grande è:", massimo)
11
12
```

The terminal on the right shows the execution of the script:

```
(kali@kali) - [~/Documents]
$ python eserBase5.py
Inserisci una lista di numeri separati da spazi: 5 8 7 3
il numero di grande è: 8.0
```

## ESERCIZIO 6 INVERTIRE UNA STRINGA

The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The Explorer panel on the left shows a directory structure with files like `1_moltiplicazione.c`, `eserBase1.py`, `eserBase2.py`, `eserBase3.py`, `eserBase4.py`, `eserBase5.py`, `eserBase6.py`, `esercizio1.py`, and `esercizio2.py`. The main editor window displays the code for `eserBase6.py`:

```
1 #chiedi una stringa all'utente
2 testo = input("Inserisci una stringa: ")
3
4 #inverti la stringa
5 invertita = testo[::-1]
6
7 print("stringa invertita:", invertita)
8
9
```

The terminal on the right shows the execution of the script:

```
(kali@kali) - [~/Documents]
$ python eserBase6.py
Inserisci una stringa: domodossola
stringa invertita: alossodomod

(kali@kali) - [~/Documents]
$ python eserBase6.py
Inserisci una stringa: angolo di bo logna
stringa invertita: angolob id ologna
a
```

## Bonus 1

The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The Explorer panel on the left shows a file tree with 'DOCUMENTS' containing '1\_moltiplicazione.c', '1\_moltiplicazione.out', 'C programmi', and several 'eserBase' files. The main editor window shows a Python script named 'esercizio1.py' with the following code:

```
1 def media_mobile(lista, n):
2     risultato = []
3
4     for i in range(len(lista)):
5
6         finestra = lista[max(0, i-n+1) : i+1]
7         media = sum(finestra) / len(finestra)
8         risultato.append(media)
9
10    return risultato
11
12    #Esempio del foglio
13    numeri = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
14    n = 3
15
16    print (media_mobile(numeri, n))
17
18
19
```

The output panel on the right shows the execution of the script, displaying the list of moving averages: [1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0].

## Bonus 2

The screenshot shows a Kali Linux virtual machine with the Visual Studio Code editor open. The Explorer panel on the left shows a file tree with 'DOCUMENTS' containing '1\_moltiplicazione.c', '1\_moltiplicazione.out', 'C programmi', and several 'eserBase' files. The main editor window shows a Python script named 'esercizio2.py' with the following code:

```
1 import re
2
3 def analisi_parole(testo):
4     #converti tutto in minuscolo
5     testo = testo.lower()
6
7     #rimuovi la punteggiatura
8     testo = re.sub(r'[a-zA-Z0-9]+', ' ', testo)
9
10    #dividi in testo le parole
11    parole = testo.split()
12
13    #dizionario per le occorrenze
14    occorrenze = {}
15
16    for parola in parole:
17        if parola in occorrenze:
18            occorrenze[parola] += 1
19        else:
20            occorrenze[parola] = 1
21
22    return occorrenze
23
24    #Esempio del foglio
25    testo = "Ciao, ciao, Come stai? Stai bene?"
26
27    print(analisi_parole(testo))
```

The output panel on the right shows the execution of the script, displaying the dictionary of word frequencies: {'ciao': 2, 'come': 1, 'stai': 2, 'bene': 1}.