



UNIVERSITÀ  
di **VERONA**

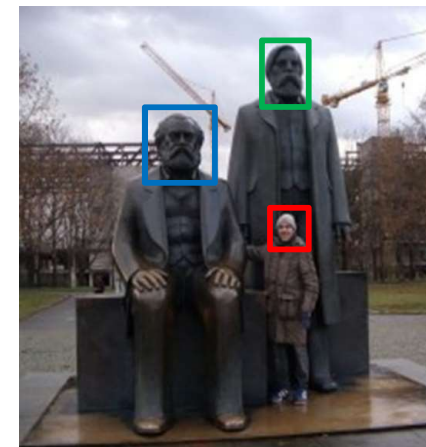
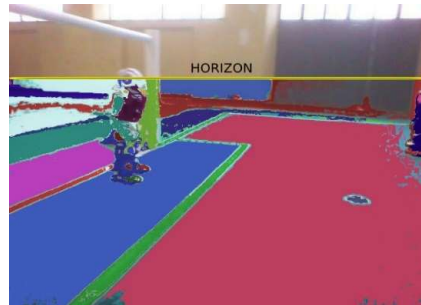
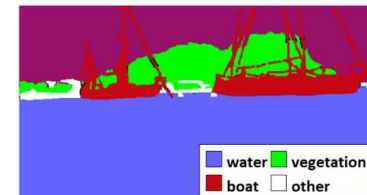
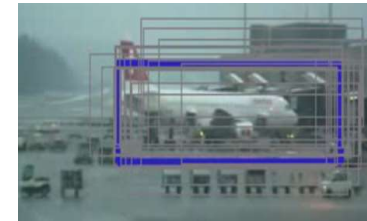
Dipartimento  
di **INFORMATICA**



# Esercizi in Python

 *Alternanza*  
SCUOLA - LAVORO


Docente:  
Domenico Daniele Bloisi




Giugno 2018

# http://profs.scienze.univr.it/~bloisi/asl/asl.html

← → ↻ ⓘ profs.scienze.univr.it/~bloisi/asl/asl.html 🔍 ☆ 📄 8 ⋮

 **Alternanza Scuola Lavoro**  
Dipartimento di Informatica  
Università di Verona



### Che cos'è l'Alternanza Scuola-Lavoro?

Un'esperienza formativa innovativa per unire sapere e saper fare, orientare le aspirazioni degli studenti e aprire didattica e apprendimento al mondo esterno. Perché l'unica risposta strutturale alla disoccupazione è una scuola collegata con il mondo del lavoro.

Attraverso il percorso di ASL si intendono promuovere le competenze chiave di cittadinanza ed in particolare:

- Agire in modo autonomo e responsabile
- Collaborare e partecipare
- Comunicare
- Acquisire ed interpretare l'informazione
- Risolvere problemi
- Imparare ad imparare

[Ministero dell'istruzione](#)

### Table of Contents

- [Lista attività](#)
- [Giugno 2018 / risultati](#)
- [Ottobre 2017 / risultati](#)
- [Luglio 2017 / risultati](#)
- [Giugno 2017 / risultati](#)

### Links

- [Home](#)
- [Università di Verona](#)
- [Prof. Paolo Fiorini](#)

### Lista attività

- [Giugno 2018](#)

# Esercizio 1

---

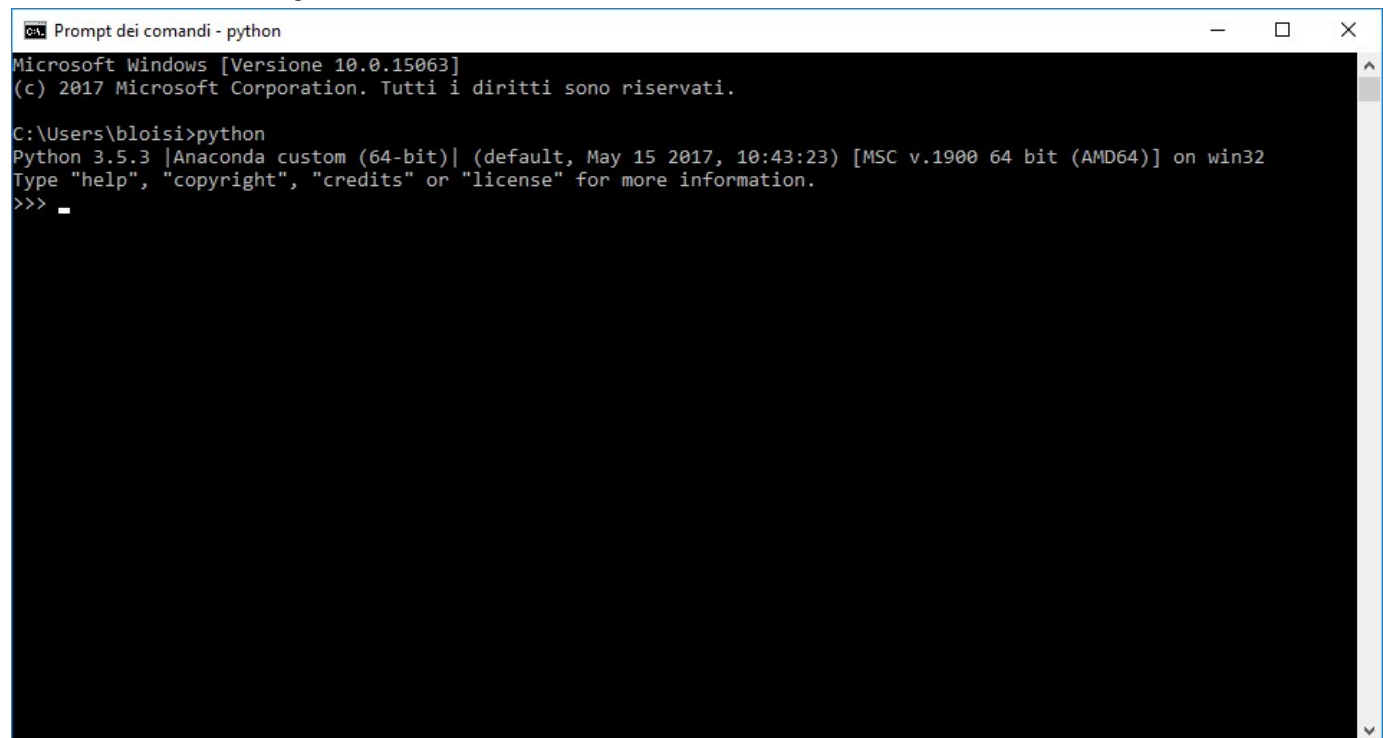
Utilizzare il linguaggio di programmazione Python per stampare a video la scritta "prima esercitazione"

# Esercizio 1 - soluzione

---

Utilizzare il linguaggio di programmazione Python per stampare a video la scritta "prima esercitazione"

Da una console  
(terminal in unix)  
digitare  
python



```
ca. Prompt dei comandi - python
Microsoft Windows [Versione 10.0.15063]
(c) 2017 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\bloisi>python
Python 3.5.3 [Anaconda custom (64-bit)] (default, May 15 2017, 10:43:23) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

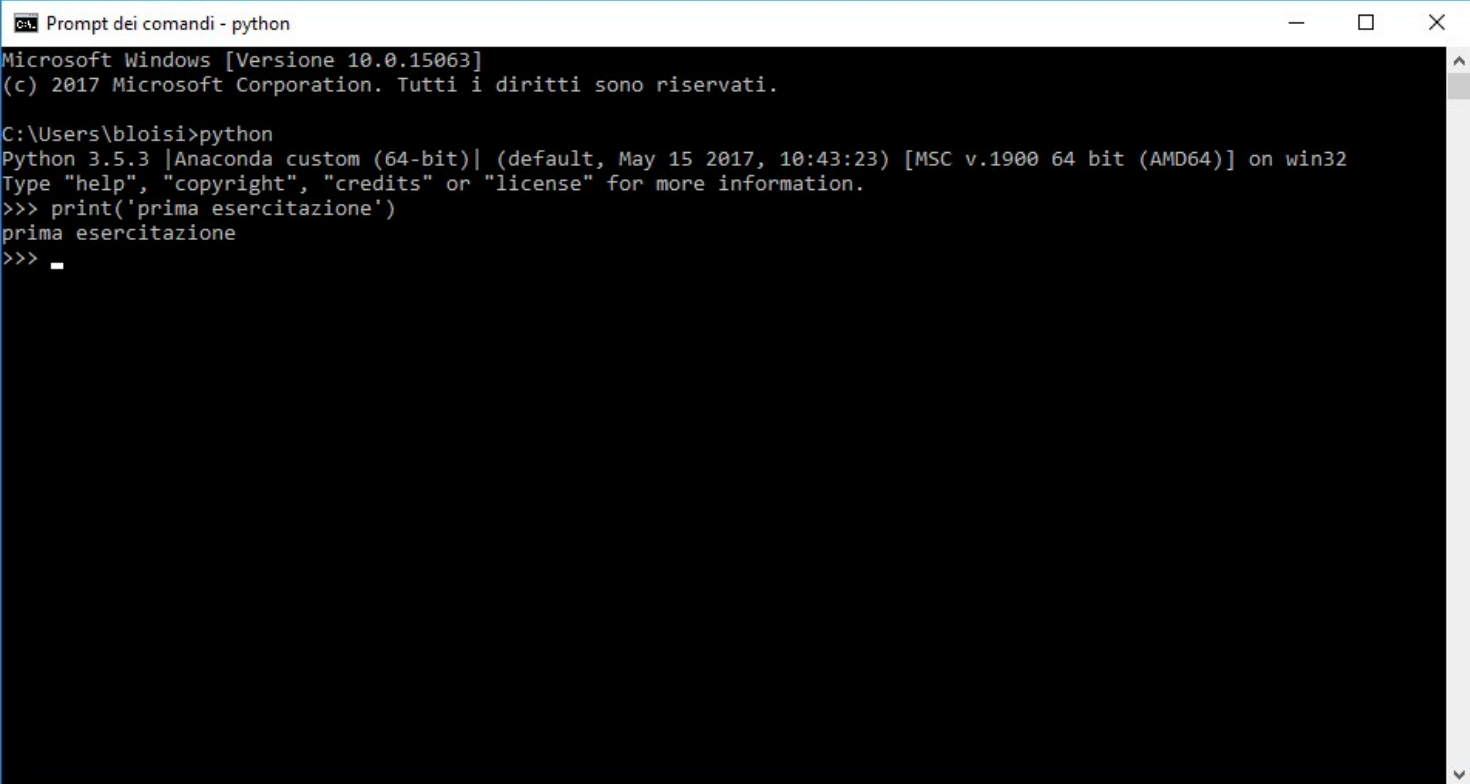
# Esercizio 1 - soluzione

---

digitare

```
print('prima esercitazione')
```

e premere invio



```
Prompt dei comandi - python
Microsoft Windows [Versione 10.0.15063]
(c) 2017 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\bloisi>python
Python 3.5.3 [Anaconda custom (64-bit)] (default, May 15 2017, 10:43:23) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('prima esercitazione')
prima esercitazione
>>> _
```

# Esercizio 2

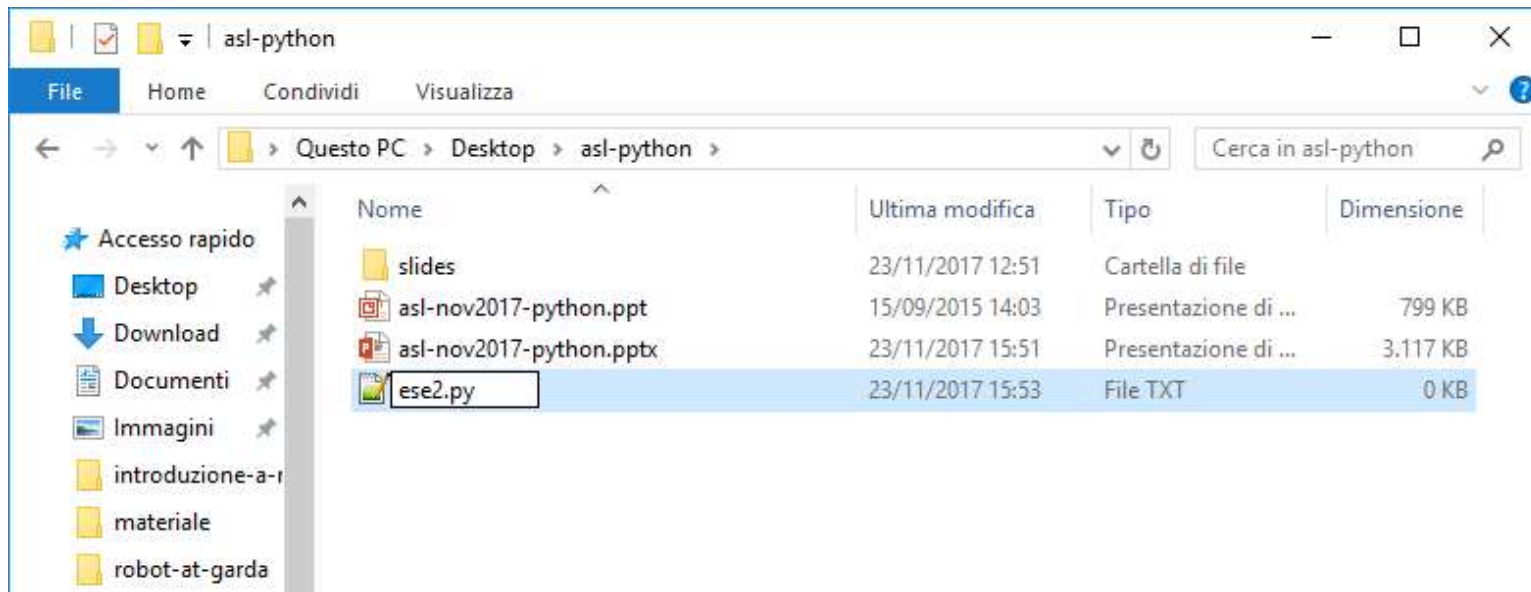
---

Ripetere l'esercizio 1 facendo in modo che le istruzioni Python siano scritte in un file di testo denominato `ese2.py`

# Esercizio 2 - soluzione

---

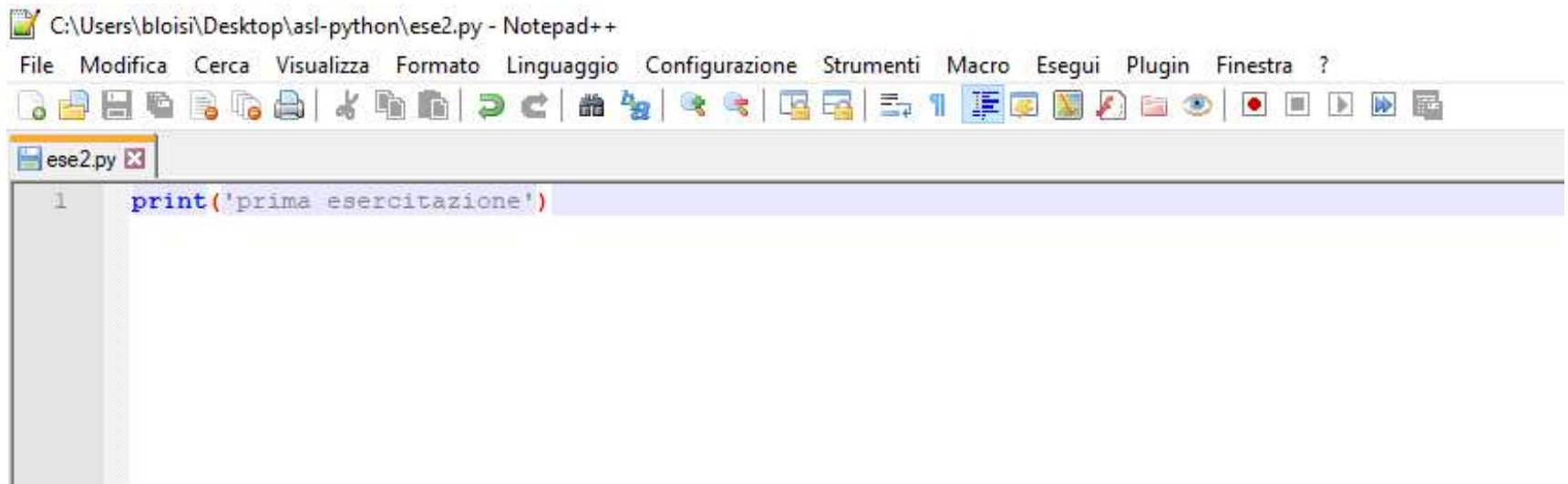
## 1. Creare il file ese2.py



# Esercizio 2 - soluzione

---

2. Aprire il file ese2.py in un editor di testo
3. Inserire il codice in linguaggio Python all'interno del file



The image shows a screenshot of the Notepad++ text editor. The title bar indicates the file path: C:\Users\bloisi\Desktop\asl-python\ese2.py - Notepad++. The menu bar includes File, Modifica, Cerca, Visualizza, Formato, Linguaggio, Configurazione, Strumenti, Macro, Esegui, Plugin, Finestra, and ?. The toolbar contains various icons for file operations and editing. The editor window shows a single line of Python code: `print('prima esercitazione')` on line 1. The code is highlighted in blue.



# Esercizio 2 - soluzione

---

4. Usando il terminal, spostarsi nella cartella che contiene il file `ese2.py` (comando `cd`)
5. digitare  
`python ese2.py`

```
C:\Users\bloisi>cd Desktop\asl-python  
  
C:\Users\bloisi\Desktop\asl-python>python ese2.py  
prima esercitazione  
  
C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 3

---

Si crei un file `ese3.py` contenente il seguente codice:

```
print('prima esercitazione')  
print('e ne seguiranno altre')
```

Che cosa si ottiene eseguendo il codice?

# Esercizio 3 - soluzione

---

Ecco cosa si ottiene eseguendo ese3.py

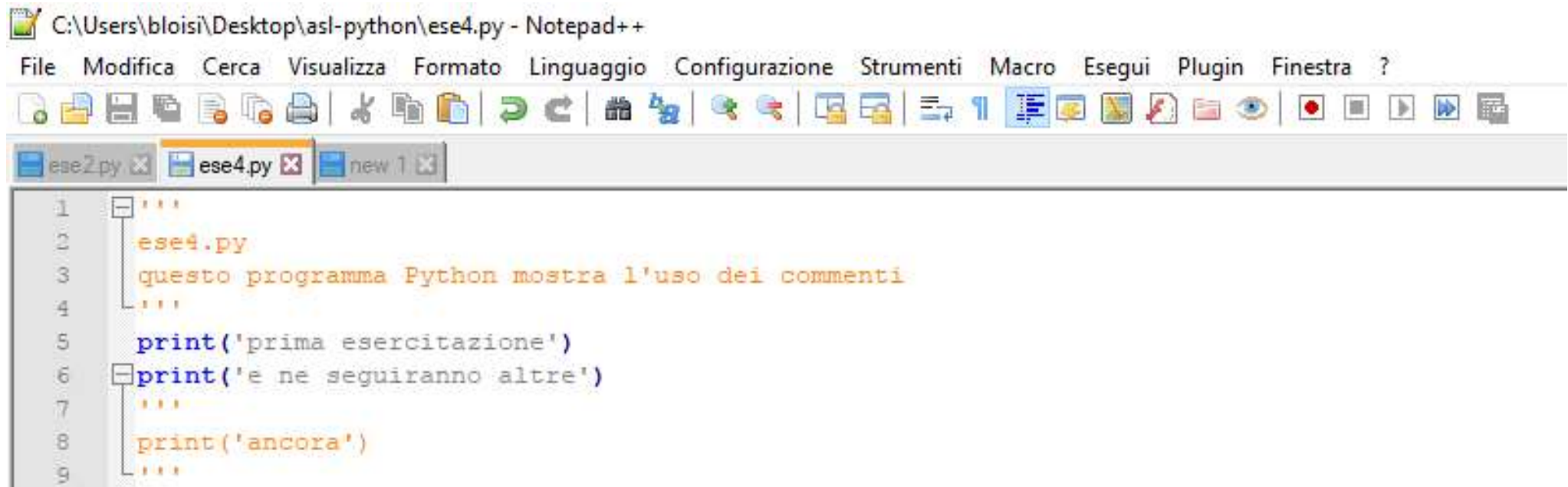
```
C:\Users\bloisi\Desktop\asl-python>python ese3.py  
prima esercitazione  
e ne seguiranno altre  
  
C:\Users\bloisi\Desktop\asl-python>_
```

Le istruzioni vengono eseguite in sequenza

# Esercizio 4

---

Si crei un file `ese4.py` come in figura



The screenshot shows the Notepad++ editor interface. The title bar reads "C:\Users\bloisi\Desktop\asl-python\ese4.py - Notepad++". The menu bar includes "File", "Modifica", "Cerca", "Visualizza", "Formato", "Linguaggio", "Configurazione", "Strumenti", "Macro", "Esegui", "Plugin", and "Finestra?". The toolbar contains various icons for file operations and editing. The editor window has three tabs: "ese2.py", "ese4.py", and "new 1". The code in the "ese4.py" tab is as follows:

```
1  """
2  ese4.py
3  questo programma Python mostra l'uso dei commenti
4  """
5  print('prima esercitazione')
6  print('e ne seguiranno altre')
7  """
8  print('ancora')
9  """
```

Che cosa si ottiene eseguendo il codice?

# Esercizio 4 - soluzione

---

Ecco cosa si ottiene eseguendo ese4.py

```
C:\Users\bloisi\Desktop\asl-python>python ese4.py  
prima esercitazione  
e ne seguiranno altre  
  
C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 5

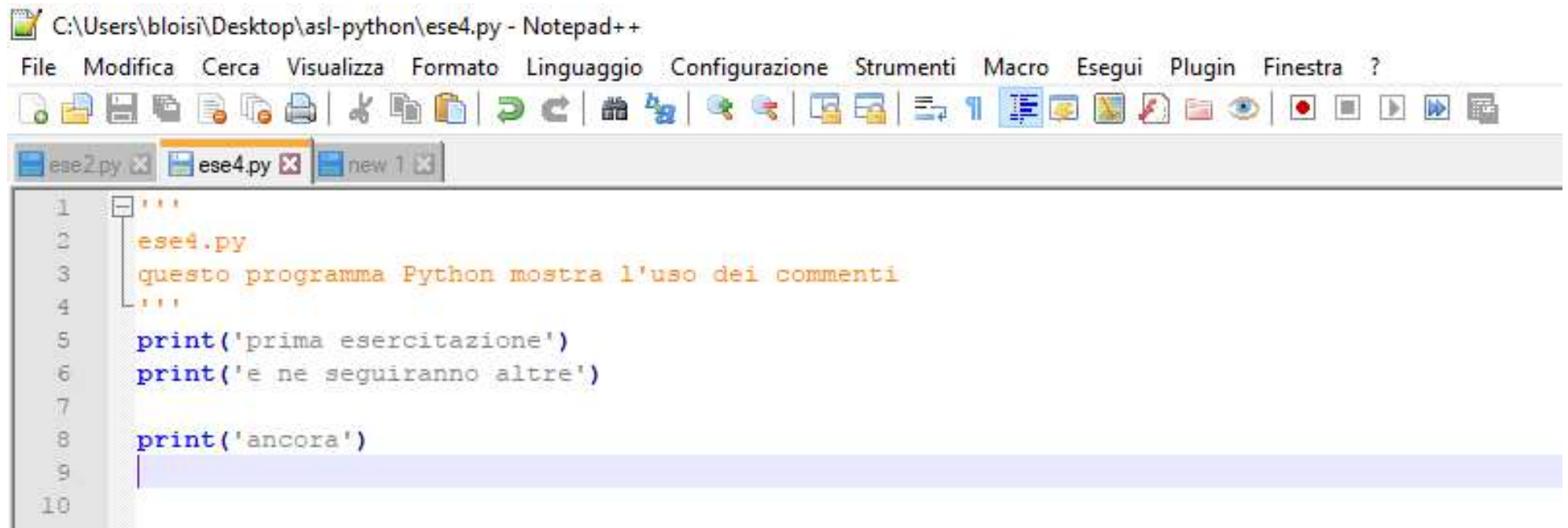
---

Modificare `ese4.py` per ottenere la stampa di

```
prima esercitazione  
e ne seguiranno altre  
ancora
```

# Esercizio 5 - soluzione

---



C:\Users\bloisi\Desktop\asl-python\ese4.py - Notepad++

File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?

ese2.py x ese4.py x new 1 x

```
1  """
2  ese4.py
3  questo programma Python mostra l'uso dei commenti
4  """
5  print('prima esercitazione')
6  print('e ne seguiranno altre')
7
8  print('ancora')
9
10
```

# Esercizio 6

---

Creare un file `ese6.py` contenente istruzioni in Python per la stampa del proprio nome, cognome, e classe nel formato mostrato sotto

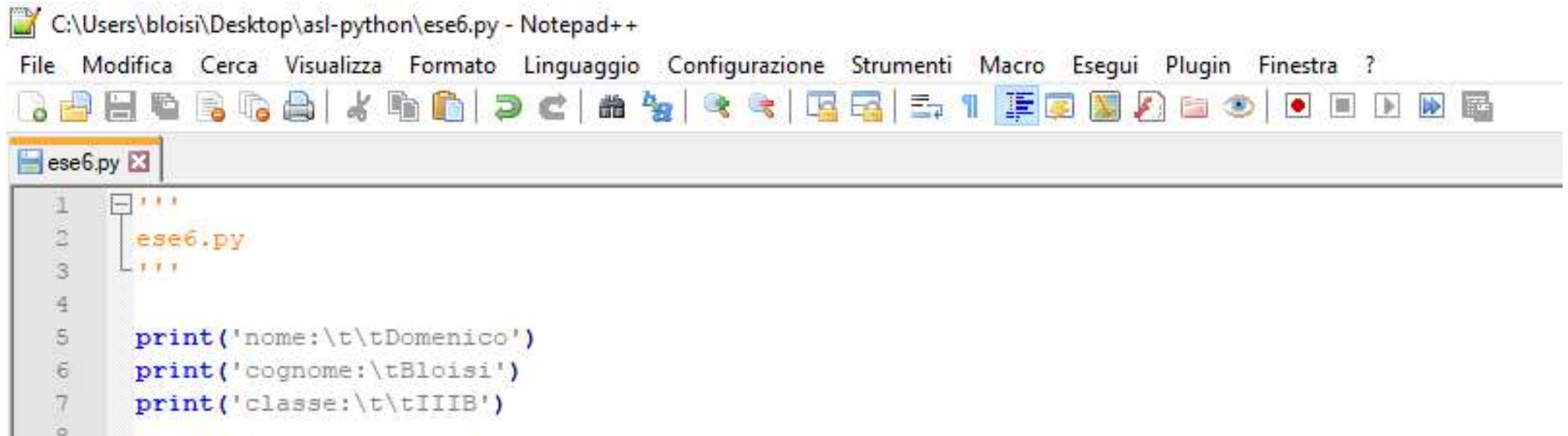
```
nome :           Domenico  
cognome :       Bloisi  
classe :        IIIB
```



# Esercizio 6 - soluzione

---

Usiamo il carattere speciale `\t` per allineare l'output su due colonne



```
C:\Users\bloisi\Desktop\asl-python\ese6.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
ese6.py x
1  """
2  ese6.py
3  """
4
5  print('nome:\t\tDomenico')
6  print('cognome:\tBloisi')
7  print('classe:\t\tIIIB')
8
```

# Esercizio 6 - soluzione

---

Ecco il risultato dell'esecuzione per ese6.py

```
C:\Users\bloisi\Desktop\asl-python>python ese6.py
nome:          Domenico
cognome:       Bloisi
classe:        IIIB

C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 7

---

Scrivere del codice Python per richiedere all'utente di inserire da tastiera il proprio nome.

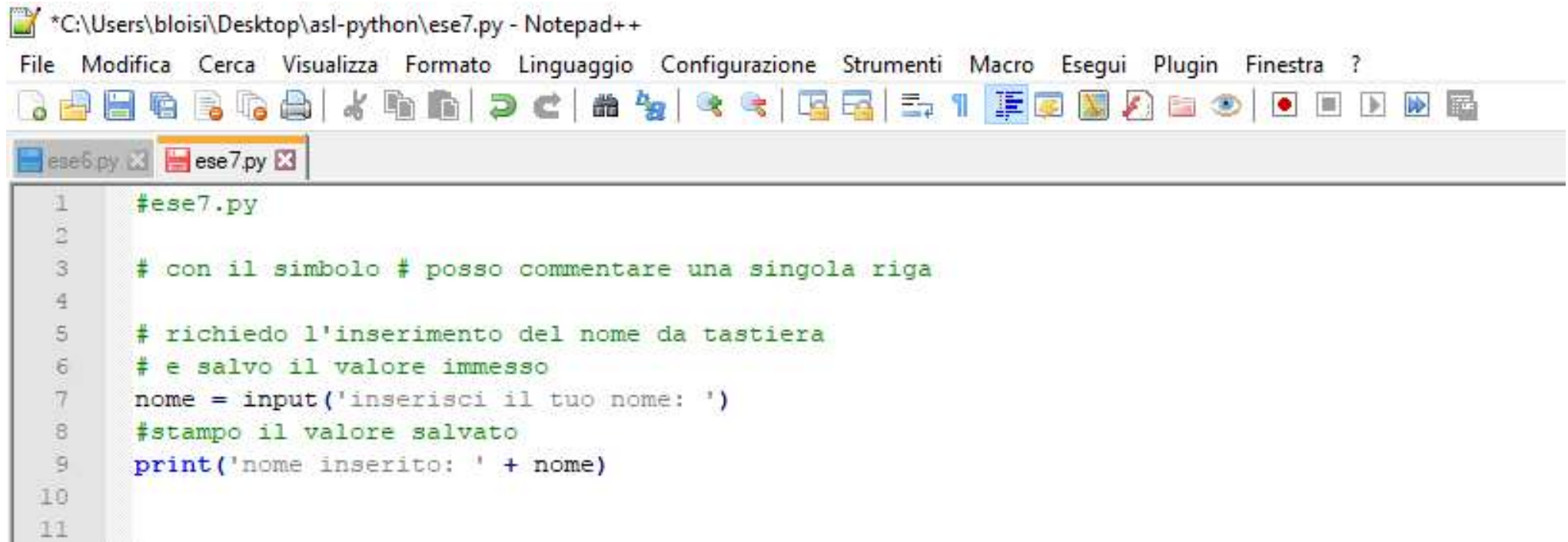
Una volta recuperato il nome, esso dovrà essere stampato a video.

Si veda l'esempio seguente:

```
inserisci il tuo nome: Domenico  
nome inserito: Domenico
```

# Esercizio 7 - soluzione

---



```
*C:\Users\bloisi\Desktop\asl-python\ese7.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
ese6.py x ese7.py x
1 #ese7.py
2
3 # con il simbolo # posso commentare una singola riga
4
5 # richiedo l'inserimento del nome da tastiera
6 # e salvo il valore immesso
7 nome = input('inserisci il tuo nome: ')
8 #stampo il valore salvato
9 print('nome inserito: ' + nome)
10
11
```

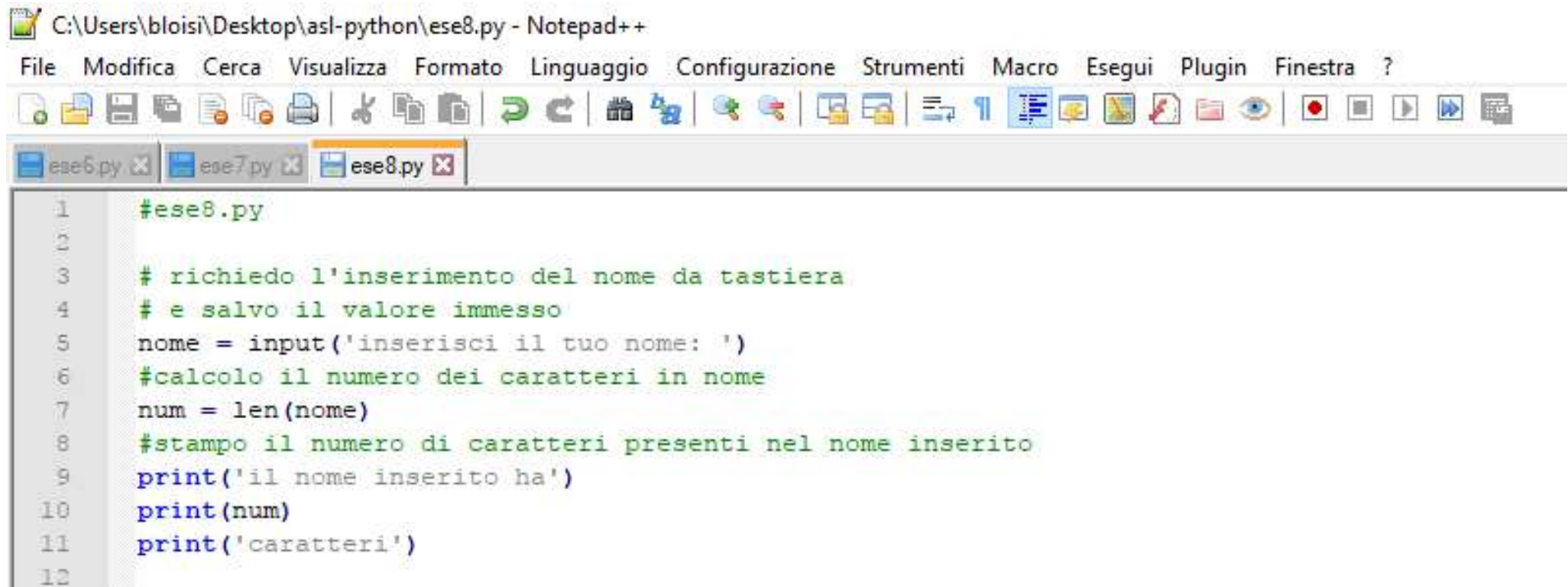
# Esercizio 8

---

Modificare `ese7.py` per stampare il numero di caratteri presenti nel nome inserito

```
inserisci il tuo nome: Domenico  
il nome inserito ha  
8  
caratteri
```

# Esercizio 8 - soluzione



```
C:\Users\bloisi\Desktop\asl-python\ese8.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
ese6.py x ese7.py x ese8.py x
1 #ese8.py
2
3 # richiedo l'inserimento del nome da tastiera
4 # e salvo il valore immesso
5 nome = input('inserisci il tuo nome: ')
6 #calcolo il numero dei caratteri in nome
7 num = len(nome)
8 #stampo il numero di caratteri presenti nel nome inserito
9 print('il nome inserito ha')
10 print(num)
11 print('caratteri')
12
```

# Esercizio 9

---

Scrivere delle istruzioni Python in grado di prendere in ingresso da tastiera un intero  $x$  e stampare a video il valore  $-x$

**Esempio 1:**

```
inserisci valore: 7
```

```
valore con segno invertito:
```

```
-7
```

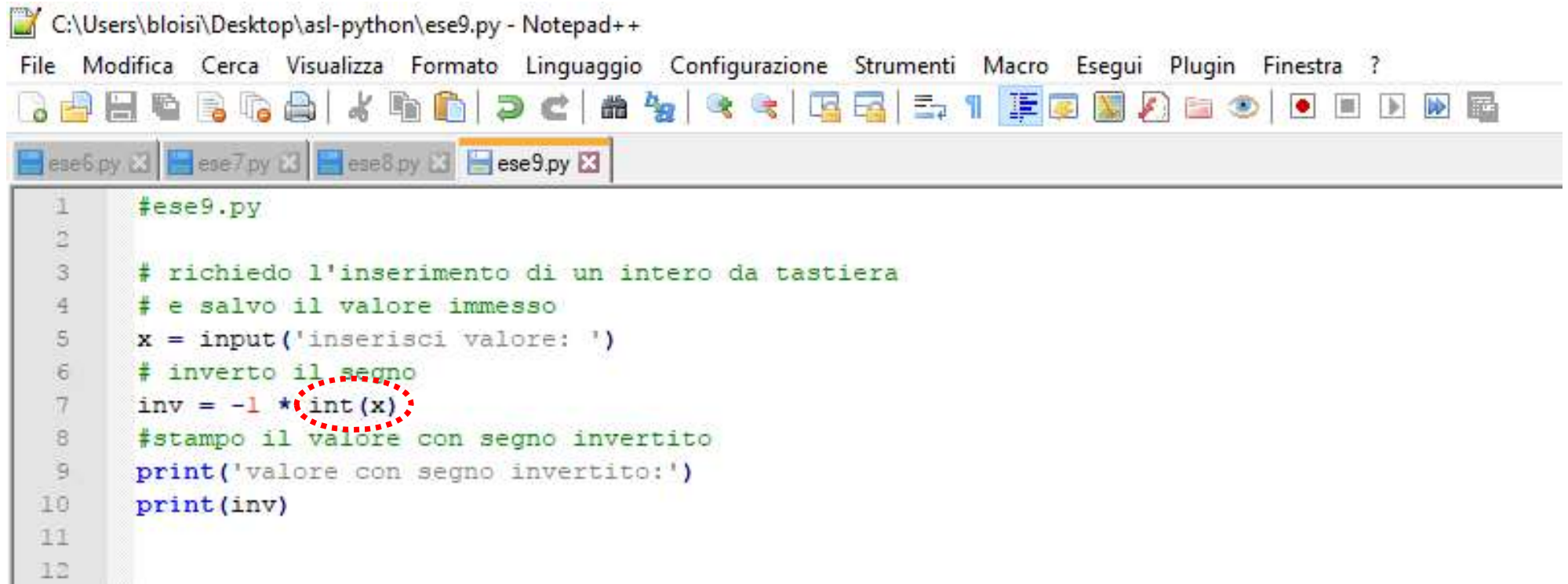
**Esempio 2:**

```
inserisci valore: -8
```

```
valore con segno invertito:
```

```
8
```

# Esercizio 9 - soluzione



The image shows a Notepad++ window with the following content:

```
C:\Users\bloisi\Desktop\asl-python\ese9.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
ese6.py x ese7.py x ese8.py x ese9.py x
1 #ese9.py
2
3 # richiedo l'inserimento di un intero da tastiera
4 # e salvo il valore immesso
5 x = input('inserisci valore: ')
6 # inverto il segno
7 inv = -1 * int(x)
8 #stampo il valore con segno invertito
9 print('valore con segno invertito:')
10 print(inv)
11
12
```



# Esercizio 10

---

Modificare ese9.py per ottenere la stampa di

```
inserisci valore: -8
```

```
valore con segno invertito: 8
```

# Esercizio 10 - soluzione

```
C:\Users\bloisi\Desktop\asl-python\ese10.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
ese6.py x ese7.py x ese8.py x ese9.py x ese10.py x new 1 x
1 #ese10.py
2
3 # richiedo l'inserimento di un intero da tastiera
4 # e salvo il valore immesso
5 x = input('inserisci valore: ')
6 # inverto il segno
7 inv = -1 * int(x)
8 #stampo il valore con segno invertito
9 print('valore con segno invertito: ' + str(inv))
10
```

# Esercizio 11

---

Si chieda all'utente di inserire due valori reali (di tipo **float**) x e y, stampando il valore  $(x+y)/(x-y)$

Esempio d'uso:

Inserire primo valore (reale):

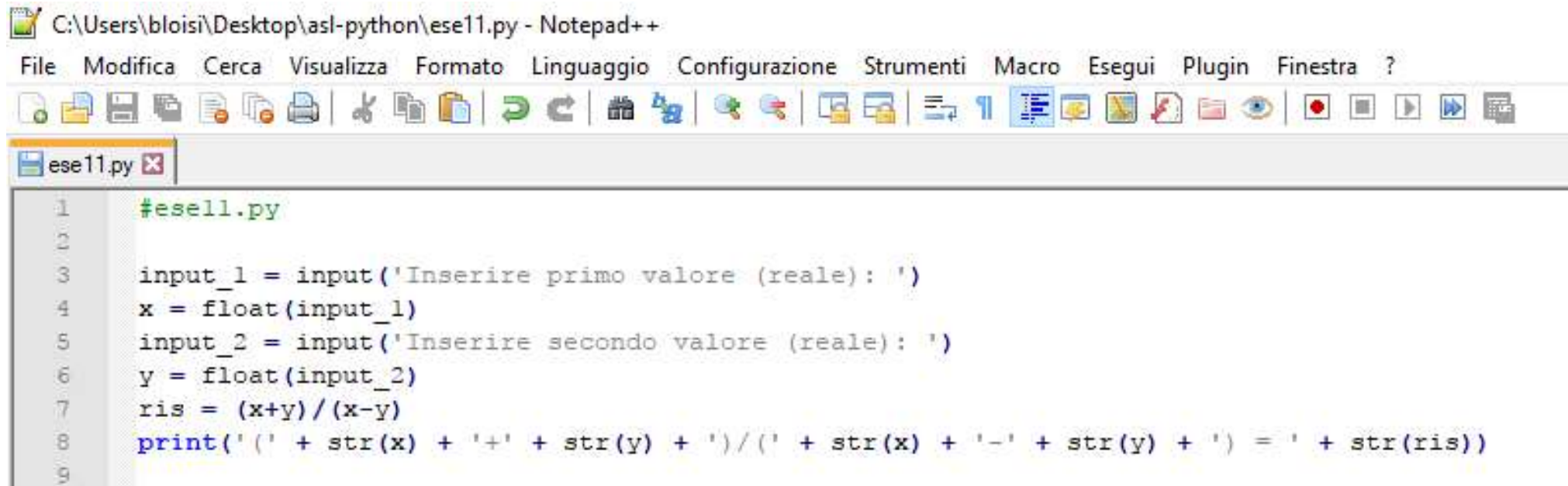
2.4

Inserire secondo valore (reale):

7.12

$(2.4+7.12) / (2.4-7.12) = -2.0169491525423724$

# Esercizio 11 - soluzione



The screenshot shows a Notepad++ window with the following content:

```
1 #esell.py
2
3 input_1 = input('Inserire primo valore (reale): ')
4 x = float(input_1)
5 input_2 = input('Inserire secondo valore (reale): ')
6 y = float(input_2)
7 ris = (x+y)/(x-y)
8 print('(' + str(x) + '+' + str(y) + ')/(' + str(x) + '-' + str(y) + ') = ' + str(ris))
9
```

$$(x + y) / (x - y)$$

# Esercizio 12

---

Scrivere del codice in Python per calcolare il numero delle ore corrispondenti all'età di una persona (espressa in anni)

In particolare, il codice deve permettere di:

1. Richiedere all'utente di inserire la propria età in anni
2. Stampare a video il numero di ore corrispondenti

Si assuma che valga sempre 1 anno = 365 giorni

```
C:\Users\bloisi\Desktop\asl-python>python ese12.py
Inserire gli anni di eta': 35
La tua eta' in ore e': 306600
C:\Users\bloisi\Desktop\asl-python>
```

 Esempio di  
esecuzione

# Esercizio 13

---

Scrivere del codice in Python per chiedere all'utente di inserire una base  $b$  e un esponente  $e$  per poi calcolare  $b^e$

```
C:\Users\bloisi\Desktop\asl-python>python ese13.py
Inserire la base: 3
Inserire l'esponente: 2
b^e: 9.0
```

```
C:\Users\bloisi\Desktop\asl-python>python ese13.py
Inserire la base: 3
Inserire l'esponente: -2
b^e: 0.1111111111111111
```

# Esercizio 13 - soluzione

```
C:\Users\bloisi\Desktop\asl-python\ese13.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro Esegui Plugin Finestra ?
ese13.py x
1 #ese13.py
2
3 import math
4
5 #chiediamo all'utente di inserire la base
6 input_1 = input('Inserire la base: ')
7 #trasformiamo i dati inseriti in interi
8 b = int(input_1)
9 #chiediamo all'utente di inserire l'esponente
10 input_2 = input('Inserire l\'esponente: ')
11 #trasformiamo i dati inseriti in interi
12 e = int(input_2)
13 #calcoliamo b^e
14 potenza = math.pow(b, e)
15 print('b^e: ' + str(potenza))
```

# Esercizio 14

---

Scrivere del codice in Python per calcolare la radice quadrata di un numero intero e  $> 0$  inserito da tastiera

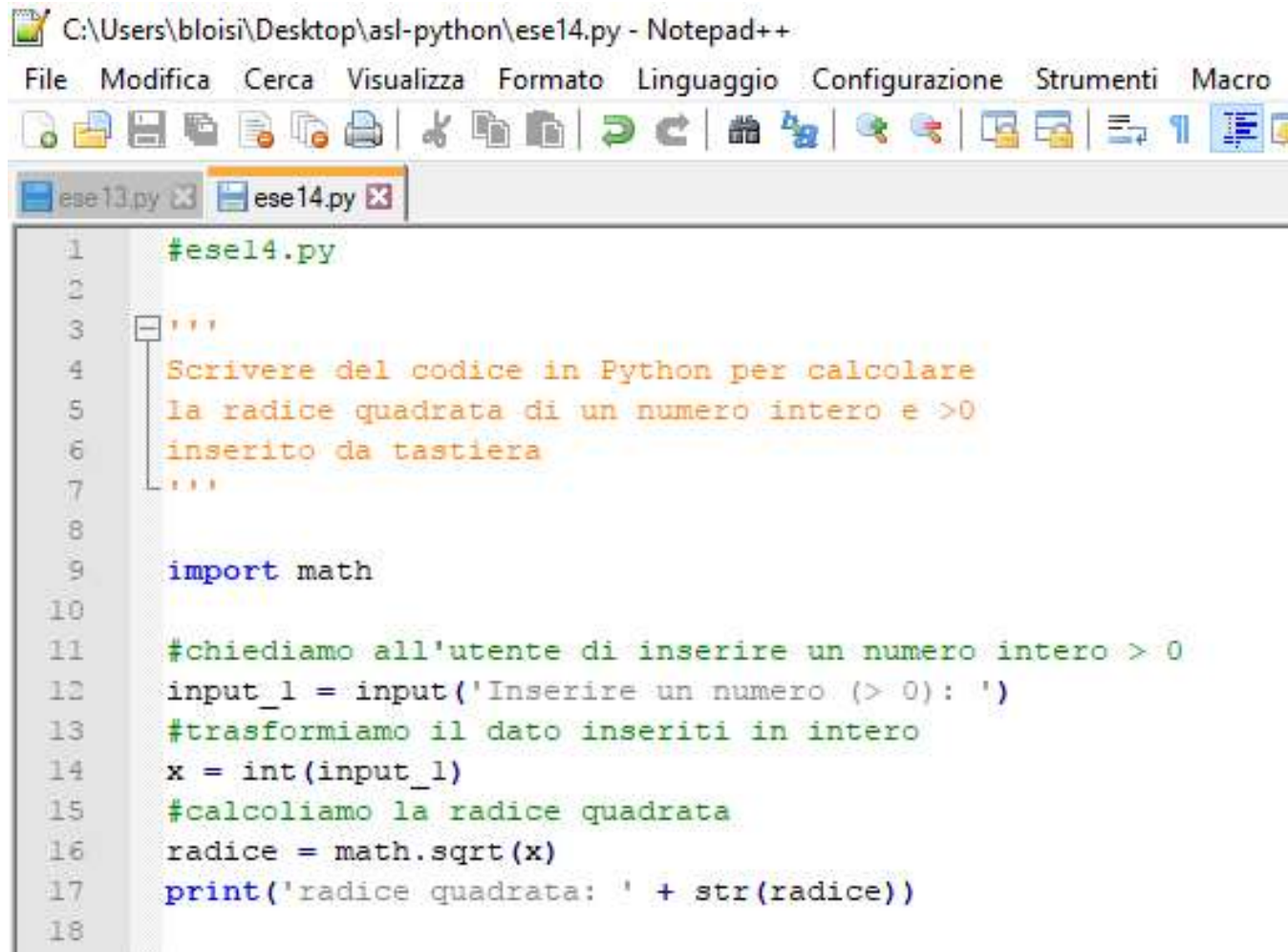
```
C:\Users\bloisi\Desktop\asl-python>python ese14.py  
Inserire un numero (> 0): 25  
radice quadrata: 5.0
```

```
C:\Users\bloisi\Desktop\asl-python>python ese14.py  
Inserire un numero (> 0): 144  
radice quadrata: 12.0
```

```
C:\Users\bloisi\Desktop\asl-python>
```



# Esercizio 14 - soluzione



```
C:\Users\bloisi\Desktop\asl-python\ese14.py - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Strumenti Macro
ese13.py x ese14.py x
1 #ese14.py
2
3 '''
4 Scrivere del codice in Python per calcolare
5 la radice quadrata di un numero intero e >0
6 inserito da tastiera
7 '''
8
9 import math
10
11 #chiediamo all'utente di inserire un numero intero > 0
12 input_1 = input('Inserire un numero (> 0): ')
13 #trasformiamo il dato inseriti in intero
14 x = int(input_1)
15 #calcoliamo la radice quadrata
16 radice = math.sqrt(x)
17 print('radice quadrata: ' + str(radice))
18
```

Cosa succede se il numero inserito da tastiera non è intero e  $> 0$ ?

# Esercizio 15

---

Scrivere un codice in Python che legga in input da tastiera le coordinate di 2 punti nel piano cartesiano  $(x_1, y_1)$  e  $(x_2, y_2)$ , calcoli la loro distanza Euclidea e la stampi a video

```
C:\Users\bloisi\Desktop\asl-python>python ese15.py
Inserire x1: 4
Inserire y1: 5
primo punto: (4, 5)
Inserire x2: 3
Inserire y2: 2
secondo punto: (3, 2)
distanza Euclidea: 3.1622776601683795
C:\Users\bloisi\Desktop\asl-python>_
```

# Esercizio 16

---

Modificare il codice soluzione dell'esercizio 14 per gestire situazioni in cui numero inserito da tastiera non sia intero e  $> 0$

```
C:\Users\bloisi\Desktop\asl-python>python ese16.py
Inserire un numero intero (> 0): 45.3
Non e' un intero!

C:\Users\bloisi\Desktop\asl-python>python ese16.py
Inserire un numero intero (> 0): 45
radice quadrata: 6.708203932499369

C:\Users\bloisi\Desktop\asl-python>python ese16.py
Inserire un numero intero (> 0): -45
Non e' > 0!

C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 16 - soluzione

```
1 #es16.py
2
3 '''
4 Scrivere del codice in Python per calcolare
5 la radice quadrata di un numero intero e >0
6 inserito da tastiera
7 '''
8
9 import math
10 import os
11
12 #chiediamo all'utente di inserire un numero intero > 0
13 input_1 = input('Inserire un numero intero (> 0): ')
14 #proviamo (try) a trasformare il dato inserito in intero
15 try:
16     x = int(input_1)
17 except ValueError:
18     print("Non e' un intero!")
19     os._exit(1)
20 #se e' intero, verificiamo che sia > 0
21 if x > 0:
22     #calcoliamo la radice quadrata
23     radice = math.sqrt(x)
24     print('radice quadrata: ' + str(radice))
25 else:
26     print("Non e' > 0!")
27
```

try  
except

if  
else

# Esercizio 17

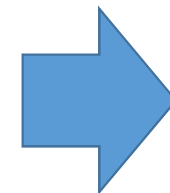
---

Scrivere un codice che legga da tastiera 3 numeri interi e stampi a video il maggiore tra essi, il minore tra essi, la media aritmetica e la radice quadrata della somma (se possibile)

```
C:\Users\bloisi\Desktop\asl-python>python ese17.py
Inserire primo numero intero: 13
Inserire secondo numero intero: -3
Inserire terzo numero intero: 15
maggiore:
15
minore:
-3
media aritmetica:
8.333333333333334
radice della somma:
5.0
C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 17 - soluzione

```
1 #es17.py
2
3 '''
4 Scrivere un codice che legga da tastiera 3 numeri interi e
5 stampi a video il maggiore tra essi, il minore tra essi,
6 la media aritmetica e la radice quadrata della somma (se possibile)
7 '''
8
9 import math
10 import os
11
12 input_1 = input('Inserire primo numero intero: ')
13 try:
14     n1 = int(input_1)
15 except ValueError:
16     print("Non e' un intero!")
17     os._exit(1)
18 input_2 = input('Inserire secondo numero intero: ')
19 try:
20     n2 = int(input_2)
21 except ValueError:
22     print("Non e' un intero!")
23     os._exit(1)
24 input_3 = input('Inserire terzo numero intero: ')
25 try:
26     n3 = int(input_3)
27 except ValueError:
28     print("Non e' un intero!")
29     os._exit(1)
30
```



# Esercizio 17 - soluzione

---

```
31 print('maggiore:')
32 if n1 > n2:
33     if n1 > n3:
34         print(str(n1))
35     else:
36         print(str(n3))
37 else:
38     if n2 > n3:
39         print(str(n2))
40     else:
41         print(str(n3))
```



# Esercizio 17 - soluzione

---

```
43     print('minore:')
44     if n1 < n2:
45         if n1 < n3:
46             print(str(n1))
47         else:
48             print(str(n3))
49     else:
50         if n2 < n3:
51             print(str(n2))
52         else:
53             print(str(n3))
54
```





# Esercizio 17 - soluzione

---

```
55     print('media aritmetica:')
56     media = (n1 + n2 + n3) / 3
57     print(str(media))
58
59     print('radice della somma:')
60     somma = n1 + n2 + n3
61     if somma > 0:
62         radice = math.sqrt(somma)
63         print(str(radice))
64     else:
65         print("La somma non e' > 0!")
```

# Esercizio 18

---

Scrivere un codice Python che legga da tastiera un valore nell'intervallo [1,12] – corrispondente ad un mese – e stampi la stagione relativa al mese inserito.

Il codice deve cercare di intercettare possibili situazioni di errore dovute a input fuori dall'intervallo predefinito.

```
C:\Users\bloisi\Desktop\asl-python>python ese18.py
Inserire un valore tra 1 e 12: 8
Estate

C:\Users\bloisi\Desktop\asl-python>python ese18.py
Inserire un valore tra 1 e 12: 15
Valore non valido!

C:\Users\bloisi\Desktop\asl-python>python ese18.py
Inserire un valore tra 1 e 12: a
Non e' un intero!

C:\Users\bloisi\Desktop\asl-python>python ese18.py
Inserire un valore tra 1 e 12: 11
Autunno

C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 18 - soluzione

---

```
12 import os
13
14 input_1 = input('Inserire un valore tra 1 e 12: ')
15 try:
16     mese = int(input_1)
17 except ValueError:
18     print("Non e' un intero!")
19     os._exit(1)
20
21 if mese > 12 or mese < 1:
22     print("Valore non valido!")
23     os._exit(1)
24
25 if mese <= 3 or mese == 12:
26     print('Inverno')
27
28 elif mese > 3 and mese <= 6:
29     print('Primavera')
30
31 elif mese > 6 and mese < 10:
32     print('Estate')
33
34 else:
35     print('Autunno')
```

Che valore viene stampato nel caso si inserisca il valore 6 (che corrisponde al mese di giugno)?  
E cosa accade inserendo 12?

# Esercizio 19

---

Modificare il codice dell'esercizio 18 in modo che si possa inserire anche il giorno, oltre al mese, per avere l'indicazione della stagione

```
C:\Users\bloisi\Desktop\asl-python>python ese19.py
Inserire mese (valore tra 1 e 12): 6
Inserire giorno del mese: 21
Estate

C:\Users\bloisi\Desktop\asl-python>python ese19.py
Inserire mese (valore tra 1 e 12): 6
Inserire giorno del mese: 20
Primavera

C:\Users\bloisi\Desktop\asl-python>python ese19.py
Inserire mese (valore tra 1 e 12): 6
Inserire giorno del mese: 32
Valore non valido!

C:\Users\bloisi\Desktop\asl-python>python ese19.py
Inserire mese (valore tra 1 e 12): -1
Valore non valido!

C:\Users\bloisi\Desktop\asl-python>python ese19.py
Inserire mese (valore tra 1 e 12): a
Non e' un intero!
```



Equinozio

<https://it.wikipedia.org/wiki/Equinozio>

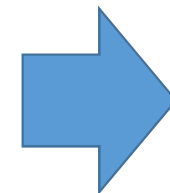
Solstizio

<https://it.wikipedia.org/wiki/Solstizio>

# Esercizio 19 - soluzione

---

```
4 import os
5
6 input_1 = input('Inserire mese (valore tra 1 e 12): ')
7 try:
8     mese = int(input_1)
9 except ValueError:
10     print("Non e' un intero!")
11     os._exit(1)
12
13 if mese > 12 or mese < 1:
14     print("Valore non valido!")
15     os._exit(1)
16
17 input_2 = input('Inserire giorno del mese: ')
18 try:
19     giorno = int(input_2)
20 except ValueError:
21     print("Non e' un intero!")
22     os._exit(1)
23
24 if giorno < 1:
25     print("Valore non valido!")
26     os._exit(1)
27
28 if (mese == 2 and giorno > 29) or \
29     ((mese == 11 or mese == 4 or mese == 6 or mese == 9) and giorno > 30) or \
30     ((mese != 2 and mese != 11 and mese != 4 and mese != 6 and mese != 9) and giorno > 31):
31     print("Valore non valido!")
32     os._exit(1)
33
```



# Esercizio 19 - soluzione

---

```
28 if (mese == 2 and giorno > 29) or \  
29     ((mese == 11 or mese == 4 or mese == 6 or mese == 9) and giorno > 30) or \  
30     ((mese != 2 and mese != 11 and mese != 4 and mese != 6 and mese != 9) and giorno > 31):  
31     ..  
32     print("Valore non valido!")  
33     os._exit(1)  
34  
35 if mese == 12:  
36     if giorno < 21:  
37         print('Autunno')  
38     else:  
39         print('Inverno')  
40  
41 elif mese < 3:  
42     print('Inverno')  
43  
44 elif mese == 3:  
45     if giorno < 21:  
46         print('Inverno')  
47     else:  
48         print('Primavera')
```



# Esercizio 19 - soluzione

---

```
49
50 elif mese < 6:
51     print('Primavera')
52
53 elif mese == 6:
54     if giorno < 21:
55         print('Primavera')
56     else:
57         print('Estate')
58
59 elif mese > 6 and mese < 10:
60     print('Estate')
61
62 else:
63     print('Autunno')
64
65
```

# Esercizio 20

---

Si scriva un codice Python che riceva come input da tastiera un intero  $n$  e disegni sullo schermo un numero di caratteri '\*' pari ad  $n$

```
C:\Users\bloisi\Desktop\asl-python>python ese20.py
valore intero n: 5
*****
finito

C:\Users\bloisi\Desktop\asl-python>python ese20.py
valore intero n: 21
*****
finito

C:\Users\bloisi\Desktop\asl-python>python ese20.py
valore intero n: Erika
Non e' un intero!

C:\Users\bloisi\Desktop\asl-python>_
```



# Esercizio 20 - soluzione

```
1 #ese20.py
2
3 import os
4
5 input_1 = input('valore intero n: ')
6 try:
7     n = int(input_1)
8 except ValueError:
9     print("Non e' un intero!")
10    os._exit(1)
11
12 i = 0
13 while i < n:
14     print('*', end='')
15     i = i + 1
16     print('\n', end='')
17
18 print('finito')
```

`print('*', end='')`

Serve per  
evitare di andare  
a capo ad ogni  
print

while condizione:  
istruzioni da  
ripetere

# Esercizio 21

---

Si modifichi il codice dell'esercizio precedente in modo da ripresentare all'utente l'inserimento di  $n$  in caso di input non numerico

```
C:\Users\bloisi\Desktop\asl-python>python ese21.py
valore intero n: 3
***
finito

C:\Users\bloisi\Desktop\asl-python>python ese21.py
valore intero n: a
a non e' un intero!
valore intero n: b
b non e' un intero!
valore intero n: 5
*****
finito

C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 21 - soluzione

```
1 #ese21.py
2
3 while True:
4     input_n = input('valore intero n: ')
5     valore_ok = True
6     try:
7         n = int(input_n)
8     except ValueError:
9         print(input_n + " non e' un intero!")
10        valore_ok = False
11    if valore_ok:
12        break
13
14    i = 0
15    while i < n:
16        print(' ', end='')
17        i = i + 1
18    print('\n', end='')
19
20    print('finito')
```

break  
serve per  
uscire dal  
ciclo while

while VERO:  
istruzioni da  
ripetere fino a  
che il valore  
immesso non  
sia ok

# Esercizio 22

---

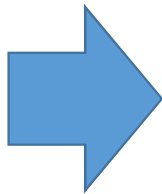
Modificare il codice dell'esercizio 21 in modo da verificare anche che  $n$  sia un valore  $> 0$

```
C:\Users\bloisi\Desktop\asl-python>python ese22.py
valore intero n: a
a non e' un intero!
valore intero n: 2.5
2.5 non e' un intero!
valore intero n: -1
-1 non e' > 0!
valore intero n: 7
*****
finito
C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 23

---

Si scriva un codice che riceva come input da tastiera due interi  $a$  e  $b$  e disegni sullo schermo un rettangolo di dimensioni  $a \times b$  usando il carattere ' \* ', così come mostrato nell'esempio



```
C:\Users\bloisi\Desktop\asl-python>python ese23.py
Lato a: 5
Lato b: 7
* * * * *
*       *
*       *
*       *
*       *
*       *
* * * * *

C:\Users\bloisi\Desktop\asl-python>python ese23.py
Lato a: 4
Lato b: 4
* * * *
*       *
*       *
* * * *

C:\Users\bloisi\Desktop\asl-python>python ese23.py
Lato a: 8
Lato b: 3
* * * * * * * *
*                   *
* * * * * * * *

C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 23 - soluzione

```
1 #ese23.py
2
3 import os
4
5 input_1 = input('Lato a: ')
6 try:
7     a = int(input_1)
8 except ValueError:
9     print("Non e' un intero!")
10    os._exit(1)
11
12 input_2 = input('Lato b: ')
13 try:
14     b = int(input_2)
15 except ValueError:
16     print("Non e' un intero!")
17    os._exit(1)
18
19 for riga in range(0, b):
20     if riga == 0 or riga == (b - 1):
21         for colonna in range(0, a):
22             print('* ', end='')
23             print('\n', end='')
24         else:
25             for colonna in range(0, a):
26                 if colonna == 0 or colonna == (a - 1):
27                     print('* ', end='')
28                 else:
29                     print(' ', end='')
30             print('\n', end='')
```

for valore in range:  
istruzioni da  
ripetere

# Esercizio 24

---

Scrivere un programma che legga da tastiera un intero  $h$  compreso tra 1 e 9 e stampi una piramide di numeri di altezza  $h$

```
C:\Users\bloisi\Desktop\asl-python>python ese24.py
Altezza: 5
  1
 121
12321
1234321
123454321
C:\Users\bloisi\Desktop\asl-python>
```

# Esercizio 24 - soluzione

```
1 #ese24.py
2
3 while True:
4     input_h = input('Altezza: ')
5     try:
6         h = int(input_h)
7     except ValueError:
8         pass
9     if h > 0 or h <= 9:
10        break;
11
12    contatore = 0
13    for riga in range (1, h + 1):
14        for colonna in range(1, h - riga + 1):
15            print(' ', end='')
16
17            contatore = 1
18            for j in range (h - riga + 1, h + 1):
19                print(str(contatore), end='')
20                contatore = contatore + 1
21            contatore = riga - 1
22            for j in range(h - riga + 2, h + 1):
23                print(str(contatore), end='')
24                contatore = contatore - 1
25    print('\n', end='')
```

Usare carta e penna per simulare l'esecuzione del codice



# Esercizio 25

---

Scrivere un codice che legga in input un valore intero  $x$  e stampi un istogramma corrispondente alle cifre di  $x$ .  
Si veda l'esempio di esecuzione

```
C:\Users\bloisi\Desktop\asl-python>cd Desktop\asl-python
Impossibile trovare il percorso specificato.

C:\Users\bloisi\Desktop\asl-python>python ese25.py
inserire un intero > 0: 1234
1 *
2 **
3 ***
4 ****

C:\Users\bloisi\Desktop\asl-python>python ese25.py
inserire un intero > 0: 74539
7 *****
4 ****
5 *****
3 ***
9 *****

C:\Users\bloisi\Desktop\asl-python>_
```

# Esercizio 25 – schema di soluzione

---

Seguire il seguente schema di soluzione:

- 1) leggere il valore  $n$  da tastiera
- 2) calcolare il numero di cifre del valore intero (sia esso  $c$ )
- 3) impostare una variabile accumulatore pari a  $n$
- 4) impostare un ciclo per  $i$  da  $c$  a  $1$ , all'interno del quale:
  - 4a) calcolare la cifra  $i$ -esima, tramite il calcolo accumulatore /  $10^{(i-1)}$
  - 4b) stampare tale cifra e il relativo numero di asterischi su una linea
  - 4c) aggiornare l'accumulatore togliendo la cifra  $i$ -esima, tramite il calcolo accumulatore %  $10^{(i-1)}$

# Esercizio 25 - soluzione

```
3 import math
4
5 while True:
6     input_n = input('inserire un intero > 0: ')
7     try:
8         n = int(input_n)
9     except ValueError:
10        print(input_n + " non e' un intero!")
11    if n > 0:
12        break;
13
14 x = n
15 c = 0
16 while x > 0:
17     x = x // 10
18     #print('x = ' + str(x))
19     c = c + 1
20     #print('c = ' + str(c))
21 acc = n;
22 i = c
23 while i > 0:
24     v = acc // math.pow(10., i-1);
25     print(str(int(v)) + ' ', end='')
26     j = 0
27     while j < v:
28         print('*', end='')
29         j = j + 1
30     print('\n', end='')
31     acc %= int(math.pow(10., i-1))
32     i = i - 1
```

$$x = x // 10$$

divisione intera



UNIVERSITÀ  
di **VERONA**

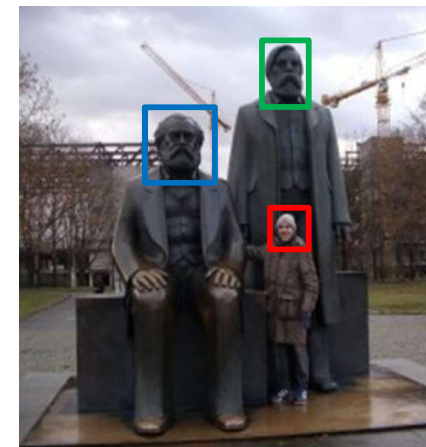
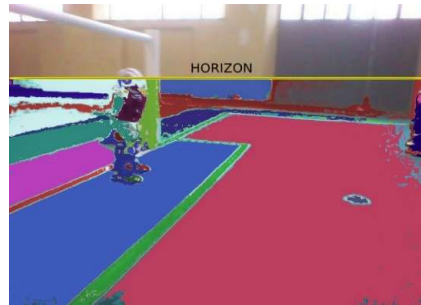
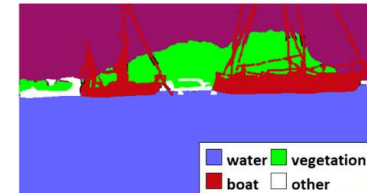
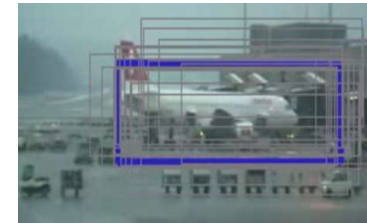
Dipartimento  
di **INFORMATICA**



# Esercizi in Python

 *Alternanza*  
SCUOLA - LAVORO

Docente:  
Domenico Daniele Bloisi



Giugno 2018