```
#PROGRAMMA 1
"""
print("PROGRAMMA1: la somma su due variabili numeriche")
a=7
b=5
c=a+b
print(c)
s = "La somma di {0} + {1} vale {2}".format(a, b, c)
print(s)
"""

"""
ESERCIZIO
Modificare il PROGRAMMA1 in tutte le sue parti in modo da eseguire il
prodotto (e non la somma) dei valori contenuti in due variabili numeriche.
Verificare che tutto funziona.
"""


#***********
#PROGRAMMA 2
"""
print("Ciao a tutti - PROGRAMMA2")
a = input("Inserisci un numero: ")
b= input("Inserisci un numero: ")
c=a+b
#print(c)
s = "La somma di {0} + {1} vale {2}".format(a, b, c)
print(s)
"""

"""
I due valori inseriti dall'utente sono memorizzati in due variabili
(var a, var b) di tipo string e quindi la somma tra due stringhe è
la concatenazione delle stringhe stesse.

ESERCIZIO

Modificare il PROGRAMMA2 in modo che il programma chiede all'utente
3 stringhe e concatena le stesse. Modificare il testo stampato a schermo
coerentemente con l'operazione fatta (no somma ma concatenazione).
Verificare che tutto funziona.
"""

#***********
#PROGRAMMA 3
"""
print("Ciao a tutti - PROGRAMMA3")
a = input("Inserisci un numero: ")
b= input("Inserisci un numero: ")
a = int(a)
b= int(b)
if(a > b):
    print("Hai inserito due numeri. Il maggiore e' ",a)
```

```
else:
    print("Hai inserito due numeri. Il maggiore e' ",b)
c=a+b
s = "La somma di {0} + {1} vale {2}".format(a, b, c)
print(s)
"""

"""
ESERCIZIO
Ripetere il programma 3 dove si chiedono all'utente 2 numeri qualsiasi e si stampa in
output la differenza tra il più grande e il più piccolo.
"""
#***********
#PROGRAMMA 4
"""
print("Ciao a tutti - PROGRAMMA 4")
sParola = input("Inserisci una parola: ")
print("Hai inserito ",sParola)
if(len(sParola) > 3):
    print('Il quarto carattere inserito è ',sParola[3])

iIndice = 0;
iTrovatoCarattere = 0
while iIndice < len(sParola):
    if(sParola[iIndice]=='a'):
        print('La parola inserita contiene la a nella posizione ',iIndice)
        iTrovatoCarattere = 1
    iIndice = iIndice + 1
if(iTrovatoCarattere==0):
    print('La parola inserita non contiene la a')
if(sParola.find('dare')>=0):
    print('La parola inserita contiene la sottostringa dare')
else:
    print('La parola inserita NON contiene la sottostringa dare')
"""
"""
ESERCIZIO
Scrivere un programma che chiede all'utente di inserire una parola
e restituisce come output la stampa di ciascuna vocale con il numero
di volte in cui la stessa compare. Es. se la parola è 'cantare'

a  2
e  1
i  0
o  0
u  0
"""


"""
ESERCIZIO
Scrivere un programma in cui si inserisce una stringa ed un carattere
ed il programma da in output il numero di volte in cui il carattere compare nella
stringa
```

```
"""
ESERCIZIO
Scrivere una semplice CALCOLATRICE cioè un programma in cui si inseriscono
due numeri ed una stringa (l'operatore) e il programma da in output il risultato.
Es.
inserisco 5
inserisco 8
inserisco x
Il programma stampa a schermo
5 x 8 = 40
"""
#PROGRAMMA 5
"""
from time import localtime

mia_agenda = {8: 'Sleeping',
              9: 'Commuting',
              17: 'Working',
              18: 'Commuting',
              20: 'Eating',
              22: 'Resting' }

print(mia_agenda[20])
time_now = localtime()
hour = time_now.tm_hour
minuti = time_now.tm_min
s = "L'ora corrente è {0}. I minuti {1}".format(hour,minuti)
print(s)
for activity_time in sorted(mia_agenda.keys()):
    if hour < activity_time:
        print (mia_agenda[activity_time])
        break
else:
    print ('Attivita sconosciuta or sleeping!')
"""
#PROGRAMMA 6
"""
print("Ciao a tutti - PROGRAMMA 6")
a = {'cane', 'Gatto', 'topo'}
print('INSIEME A:',a)
b = set()
b.add('leone')
b.add('Gatto')
b.add('tigre')
print('INSIEME B:',b)
print('UNIONE:', a | b) # unione
print('INTERSEZIONE:', a & b) # intersezione
print('DIFFERENZA:',a - b) # differenza
print('DIFFERENZA SIMMETRICA:',a ^ b) # differenza simmetrica
"""
```

```
"""
ESERCIZIO
Prova ad inserire più volte lo stesso elemento (es. 'cane')
in una variabile di tipo set. Cosa succede? Gli inserimenti
successivi al primo che effetto hanno?
"""
"""
ESERCIZIO
Definisci una variabile lista
mylist = []
e usando mylist.append("cane") prova a fare lo stesso esercizio fatto in
precedenza (inserimento dello stesso elemento più volte).
Che succede?
"""
"""
ESERCIZIO
Tenendo ben presente quanto appreso nei due esercizio precedenti
scrivere un programma che carica una lista con 10 elementi con delle
ridondanze. Utilizzare una variabile di tipo set per ottenere
l'insieme degli elementi DISTINTI della lista.

Procedura: fare un ciclo (FOR oppure WHILE) con il quale si scandisce tutta la lista.
Per ogni elemento della lista fare la ADD nella variabile set.
"""
#PROGRAMMA 7
"""
print("Ciao a tutti - PROGRAMMA 7")
a = {'cane', 'Gatto', 'topo'}
for animale in a:
    print(animale)

esami = {'IN110':27, 'AM110':30, 'AL110':24}
print("Esami superati:", end=" ")   #end=" " significa che non va a capo ma lascia uno
spazio alla fine
for corso in esami.keys():
    print(corso, end=" ")

print()
print("Valutazioni per gli esami:")
for corso in esami.keys():
    print("Corso ", corso, " voto ", esami[corso], "/30")

#esami['IN110'] VALE 27
"""
#PROGRAMMA 8
"""
def prima_procedura():
    print("Ciao, sono la prima procedura")
    return

print("Qui inizia lesecuzione del programma.")
prima_procedura()
print("Fine del programma.")
"""
```

```
#PROGRAMMA 9
"""
def prima_procedura():
    print("Ciao, sono la prima procedura")
    return

def seconda_procedura():
    print("Ciao, sono la seconda procedura")
    return

print("Qui inizia lesecuzione del programma.")
prima_procedura()
print("Fine del programma.")
"""

#PROGRAMMA 10
"""
def prima_procedura():
    seconda_procedura()
    print("Ciao, sono la prima procedura")
    return

def seconda_procedura():
    print("Ciao, sono la seconda procedura")
    return

print("Qui inizia lesecuzione del programma.")
prima_procedura()
print("Fine del programma.")
"""


"""
ESERCIZIO
Riscrivi la tua calcolatrice in cui la prima domanda che fai è l'operatore.
Hai tante procedure per quanti sono gli operatori.
LE richieste dei numeri si fanno all'interno di queste procedure.
In questo modo puoi gestire operazioni che richiedono un solo numero (es. il quadrato
di un numero) oppure operazioni che richiedono anche 10 numeri:
esempio la media aritmetica.
"""
#PROGRAMMA 11
"""
def mcm(x, y):
    mx = x
    my = y
    while mx != my:
        if mx<my:
            mx = mx+x
        else:
            my = my+y
    return(mx)

print("Ciao a tutti - PROGRAMMA6")
```

```
iPos = 0
while iPos < 3:
    sStringaRicevuta = input("Inserisci un valore per a: ")
    try:
        a = float(sStringaRicevuta)
        print("Hai inserito ",a)
        iPos = 100
    except:
        print("Attenzione, hai inserito un valore sbagliato ",sStringaRicevuta)
        iPos = iPos + 1

if(iPos!=100):
    exit(0)

b = int(input("Inserisci un valore per b: "))
m = mcm(a,b)
print("mcm(", a, ",", b,") = ", m)
"""


"""
ESERCIZIO
Aggiungi la funzionalità di Minimo comune multiplo alla tua calcolatrice.
Cerca la relazione tra Minimo comune multiplo e massimo comun divisore.
Aggiungi la funzionalità di Massimo comun divisore alla tua calcolatrice.
"""


#PROGRAMMA 12 (SOLO IN LOCALE, NO ON-LINE)
"""
import fnmatch
import os

rootPath = './'
pattern = '*.pdf'

for root, dirs, files in os.walk(rootPath):
    for filename in fnmatch.filter(files, pattern):
        print( os.path.join(root, filename))
"""


#PROGRAMMA 13 (SOLO IN LOCALE, NO ON-LINE)
"""
import fnmatch
import os

images = ['*.jpg', '*.jpeg', '*.png', '*.tif', '*.tiff']
matches = []

for root, dirnames, filenames in os.walk("C:\"):
    for extensions in images:
        for filename in fnmatch.filter(filenames, extensions):
```

```
            matches.append(os.path.join(root, filename))
"""


#crea una directory se non esiste (SOLO IN LOCALE, NO ON-LINE)
"""
import os
os.makedirs(path, exist_ok=True)

"""
#PROGRAMMA 14 (SOLO IN LOCALE, NO ON-LINE)
"""
import shutil

original = r'C:\\Users\\Ron\\Desktop\\Test_1\\image.jpg'
target = r'C:\\Users\Ron\\Desktop\\Test_2\\image.jpg'

shutil.copyfile(original, target)
"""

#ESERCIZIO (SOLO IN LOCALE, NO ON-LINE)
"""
ESERCIZIO
Chiedi all'utente di inserire un path assoluto (es. c:\\users\\mario)
Dopodiche cerca tutti i file immagine in quella cartella e nelle sue sottocartelle
e copiali in una cartella di OUTPUT che ti ha indicato l'utente stesso.
"""

Test_01_Etivity1: questo testo è un flusso di speranze.

#PROGRAMMA 15
"""
# split this string on the "newline" character.
data = """
0.000000    -2.6881
3.336670    -4.3010
6.673340    -5.3763
1.001001    -6.9892
1.334668    -1.1290
1.668335    -1.4516
2.002002    -2.0430
2.335669    -2.5268
2.669336    -3.1182
3.003003    -3.8709
3.336670    -4.6236
3.670337    -5.4300
4.004004    -6.2365
4.337671    -7.1505
4.671338    -8.0107
5.005005    -8.9246
5.338672    -9.8924
5.672339    -1.0806
6.006006    -1.1774
6.339673    -1.2741
```

```
6.673340    -1.3709
7.007007    -1.4677
7.340674    -1.5752
7.674341    -1.6720
8.008008    -1.7688
8.341675    -1.8655
8.675342    -1.9731
9.009009    -2.0752
9.342676    -2.1827
9.676343    -2.2849
""".split('\n')

print(len(data))


#
# Abbiamo una lista di stringhe. Dobbiamo prendere ciascuna stringa
# e tirarne fuori una coppia di numeri decimali (float).
#

tlist = []
ylist = []
for s in data:
    if s:
        t,y = s.split()     # split the string in two
        t=float(t)          # convert time to float
        y=float(y)/100.0    # convert y-position (cm) to float in meters
        tlist.append(t)     # append to the list for time data
        ylist.append(y)     # append to the list for y-position data

#print "tlist=",tlist
#print "ylist=",ylist

import matplotlib.pyplot as plt
plt.title('y-position vs. time for falling cupcake paper')
plt.xlabel('t (s)')
plt.ylabel('y (m)')
plt.plot(tlist,ylist,'m.')
plt.show()

"""

#PROGRAMMA 16 (SOLO IN LOCALE)
"""
https://stackoverflow.com/questions/17098675/searching-text-in-a-pdf-using-python

import PyPDF2
import re

pattern = input("Enter string pattern to search: ")
fileName = input("Enter file path and name: ")

object = PyPDF2.PdfFileReader(fileName)
numPages = object.getNumPages()
```

```
for i in range(0, numPages):
    pageObj = object.getPage(i)
    text = pageObj.extractText()

    for match in re.finditer(pattern, text):
        print(f'Page no: {i} | Match: {match}')
"""


#PROGRAMMA 17(IN LOCALE OPPURE ON LINE DOPO OPPORTUNE MODIFICHE, VEDI ESERCIZIO
SEGUENTE)
"""
Il programma stampa i 150 fiori nel piano (SepalLengthCm,PetalLengthCm)
colorando i fiori in modo diverso a seconda della specie.

https://trinket.io/embed/python3/a5bd54189b  (per usare python online con matplotlib)
"""
"""
import pandas as pd
import matplotlib.pyplot as plt
import sys

##axis=1 significa che il taglio e per colonna
df = pd.read_csv('./iris.csv')
df = df.drop(['Id'],axis=1)

target = df['Species']

#Uso la struttura dati 'insieme' perchè mi elimina le ridondanze
#In target ho 150 elementi, in s ho solo 3 elementi
s = set()
for val in target:
    s.add(val)
print(s)
#sys.exit(0)

s = list(s)
#print(s)
#print(s[0])
#print(len(s))

##df_singola_specie = df[df.Species==s[0]]
##print(df_singola_specie)

plt.figure(figsize=(8,6))

marker_vect = ['+','_','*']
color_vect = ['green','red', 'blue']
for numero in range(len(s)):
    print(numero)
    df_singola_specie = df[df.Species==s[numero]]
    x = df_singola_specie['SepalLengthCm']  #prendi tutte le x di una specie (numero)
    y = df_singola_specie['PetalLengthCm']  #prendi tutte le y di una specie (numero)
```

```
    plt.scatter(x,y,marker=marker_vect[numero],color=color_vect[numero])

plt.show()
"""

#ESERCIZIO 6
"""
ESERCIZIO
Ripeti l'esercizio precedente scegliendo due variabili diverse per gli assi
cartesiani.
Es. SepalWidth e PetalWidth oppure PetalLEngth e PetalWidth
"""

Test_02_Etivity1: questo testo è un flusso di speranze.



#ESERCIZIO 7
"""
ESERCIZIO
Ripetere il programma 17 leggendo i dati da una stringa anziché da un file CSV
dopo aver compreso bene il programma 15
"""
```