

Primo_Programma (1)-2

November 10, 2023

1 IL PRIMO PROGRAMMA

1.1 SALUTO AL MONDO

```
[1]: print ("Ciao, mondo!")
```

Ciao, mondo!

1.2 SALUTO PERSONALIZZATO

```
[4]: nome = input ("Inserisci il tuo nome: ")  
print ("Ciao,", nome, "!")
```

Inserisci il tuo nome: Matias

Ciao, Matias !

1.3 STAMPA DEL NOME

```
[20]: nome="Mattia"  
print(nome)
```

Mattia

1.4 INSERIMENTO NOME DELLA VIA

```
[1]: via=input("Inserisci nome della via ")  
print("hai inserito", via)
```

Inserisci nome della via Roma

hai inserito Roma

1.5 ITERAZIONE DI SALUTO CON IL TUO NOME

```
[2]: nome = input("inserisci il tuo nome:")  
for i in range(10):  
    print("Ciao",nome,"!") #Python ripete solo la parte di codice identata  
    →(ovvero quello che è spostato a destra)
```

```
inserisci il tuo nome:Matias
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
Ciao Matias !
```

2 OPERAZIONE MATEMATICA

2.1 SOMMA

```
[3]: numero1 = int(input("Inserisci il primo numero: "))#Int vuol dire numero intero
      numero2 = int(input("Inserisci il secondo numero: "))
      somma = numero1 + numero2
      print("La somma è:", somma)
```

```
Inserisci il primo numero: 5
Inserisci il secondo numero: 4
La somma è: 9
```

2.2 SOTTRAZIONE

```
[4]: sottrazione = numero1 - numero2
      print("La sottrazione è:", int(sottrazione))
```

```
La sottrazione è: 1
```

2.3 MOLTIPLICAZIONE

```
[5]: # Chiede all'utente di inserire il primo numero
      numero1 = int(input("Inserisci il primo numero: "))

      # Chiede all'utente di inserire il secondo numero
      numero2 = int(input("Inserisci il secondo numero: "))

      # Esegue la moltiplicazione dei due numeri
      moltiplicazione = numero1 * numero2

      # Stampa il risultato della moltiplicazione
      print("La moltiplicazione di", numero1, "e", numero2, "è:", moltiplicazione)
```

```
Inserisci il primo numero: 34
Inserisci il secondo numero: 45
La moltiplicazione è: 1530
```

2.4 DIVISIONE

```
[6]: divisione= numero1 / numero2
print("La divisione è:",divisione)
```

La divisione è: 0.7555555555555555

2.5 LOOP E RIPETIZIONE

```
[11]: # Utilizza un ciclo for per iterare attraverso i numeri da 1 a 10
for numero in range(1, 11):
    # Stampa il numero corrente
    print(numero)
```

1
2
3
4
5
6
7
8
9
10

2.6 CONDIZIONI E DECISIONI

```
[12]: # Calcolatrice di python
operazione = input("Inserisci l'operazione (+,-,*,/): ")
numero1 = float(input("Inserisci il primo numero: ")) #float con virgola
numero2 = float(input("Inserisci il secondo numero: "))
# if due uguale serve per cercare quale determinata simbolo in questo caso

if operazione == "+": #== compara il valore di due oggetti
    risultato = numero1 + numero2
elif operazione == "-":
    risultato = numero1 - numero2
elif operazione == "*":
    risultato = numero1 * numero2
elif operazione == "/":
    risultato = numero1 / numero2
else:
    risultato = "Operazione non valida"
print("Il risultato è:",risultato)
```

Inserisci l'operazione (+,-,*,/): +
Inserisci il primo numero: 65
Inserisci il secondo numero: 65
Il risultato è: 130.0

2.7 CONFRONTA FINO “N”

```
[14]: # Chiede all'utente di inserire un numero intero positivo
n = int(input("Inserisci un numero intero positivo:"))

# Utilizza un ciclo for per stampare i numeri da 1 a n
for numero in range(1, n + 1):
    # Stampa il numero corrente
    print(numero)
```

Inserisci un numero intero positivo:3

1
2
3

2.8 SOMMA NUMERI INTERI

```
[21]: #Somma
n = int(input("Inserisci un numero intero positivo:"))
somma = 0
for numero in range(1, n+1):
    somma += numero #(somma = somma + numero)
print("La somma dei primi", n, "numeri interi è:", somma)
```

Inserisci un numero intero positivo:2

La somma dei primi 2 numeri interi è: 3

2.9 CALCOLARE I QUADRATI

```
[23]: #Calcolare il quadrato dei primi numeri
n = int(input("Inserisci un numero intero positivi: "))
print("Quadrati dei primi", n,"numeri:")

for numero in range(1, n + 1):
    quadrato = numero ** 2
    print("Il quadrato di", numero,"è", quadrato)
```

Inserisci un numero intero positivi: 3

Quadrati dei primi 3 numeri:

Il quadrato di 3 è 9

2.10 VERIFICA LA PARITÀ

```
[24]: numero = int(input("Inserisci un numero: ")) #Chiede all'utente di inserire un
    ↳numero e lo memorizza in 'numero'

if numero % 2 == 0: # Controlla se 'numero' è divisibile per 2 senza resto
    print(numero, "è un numero pari.") # Se il resto della divisione per 2 è 0,
    ↳allora 'numero' è pari
```

```
else:
    print(numero, "è un numero dispari.") # Altrimenti, 'numero' è dispari
```

Inserisci un numero:4

4 è un numero pari.

2.11 FATTORIALE

```
[26]: # Chiede all'utente di inserire un numero intero positivo
n = int(input("Inserisci numero intero positivo: "))

# Inizializza il fattoriale a 1, poiché moltiplicare per 1 non cambierà il
↳risultato
fattoriale = 1

# Calcola il fattoriale utilizzando un ciclo for
for numero in range(1, n + 1):
    # Moltiplica il fattoriale per il numero corrente
    fattoriale *= numero

# Stampa il risultato del fattoriale
print("Il fattoriale di", n, "è:", fattoriale)
```

Inserisci numero intero positivo: 3

Il fattoriale di 3 è: 6

2.12 CALCOLARE LA MEDIA DI UNA LISTA DI NUMERI

```
[5]: # Inizializza una lista vuota per contenere i numeri
numeri = []

# Chiede all'utente quanti numeri vuole inserire
n = int(input("Quanti numeri vuoi inserire? "))

# Ciclo per ottenere i numeri dall'utente e aggiungerli alla lista
for i in range(n):
    numero = float(input("Inserisci un numero: ")) # float = floating point
↳punti e virgola movibili
    numeri.append(numero)

# Calcola la media dei numeri nella lista
media = sum(numeri) / len(numeri) # len = lunghezza, sum = somma

# Stampa la media e la lista completa dei numeri
print("La media dei numeri inseriti è:", media, "la lista completa è:", numeri)
```

Quanti numeri vuoi inserire? 2

Inserisci un numero: 3

Inserisci un numero: 4

La media dei numeri inseriti è: 3.5 la lista completa è: [3.0, 4.0]

2.13 INDOVINA IL NUMERO

```
[1]: # Importa il modulo random per generare numeri casuali
import random

# Genera un numero casuale da indovinare compreso tra 1 e 100
numero_da_indovinare = random.randint(1, 100) # randint = numero random
↳ compreso tra i due estremi
tentativi = 0

# Ciclo while per continuare a chiedere all'utente di indovinare
while True:
    # Richiede all'utente di fare un tentativo
    tentativo = int(input("Indovina il numero (1-100): "))
    tentativi += 1

    # Verifica se il tentativo è corretto, troppo grande o troppo piccolo
    if tentativo == numero_da_indovinare:
        print("Bravo! Hai indovinato il numero", numero_da_indovinare, "in",
↳ tentativi, "tentativi.")
        break # Interrompe il ciclo while se il numero è stato indovinato
    elif tentativo < numero_da_indovinare:
        print("Il numero è più grande.")
    else:
        print("Il numero è più piccolo.")
```

Indovina il numero (1-100): 67

Il numero è più grande.

Indovina il numero (1-100): 78

Il numero è più grande.

Indovina il numero (1-100): 89

Bravo! Hai indovinato il numero 89 in 3 tentativi.

2.14 CARTA FORBICE SASSO

```
[6]: # Importa il modulo random per la scelta casuale del computer
import random

# Definisce le possibili mosse nel gioco
mosse = ["carta", "forbici", "sasso"]

# Il computer fa una scelta casuale tra le mosse possibili
computer_mossa = random.choice(mosse)

# Stampa un messaggio di benvenuto
```

```

print("Benvenuti al Gioco della Morra Cinese!")

# Chiede all'utente di scegliere una mossa
scelta_giocatore = input("Scegli la tua mossa (carta, forbici, sasso): ")

# Verifica se la mossa dell'utente è permessa
if scelta_giocatore not in mosse:
    print("Mossa non permessa")
else:
    # Stampa la mossa del computer
    print("Il computer ha scelto:", computer_mossa)

    # Confronta le mosse dell'utente e del computer per determinare il risultato
    if scelta_giocatore == computer_mossa:
        print("Pareggio")
    elif (scelta_giocatore == "carta" and computer_mossa == "sasso") or \
          (scelta_giocatore == "forbici" and computer_mossa == "carta") or \
          (scelta_giocatore == "sasso" and computer_mossa == "forbici"):
        print("Hai vinto")
    else:
        print("Hai perso!")

```

```

Benvenuti al Gioco della Morra Cinese!
Scegli la tua mossa (carta, forbici, sasso): carta
Il computer ha scelto: sasso
Hai vinto

```

2.15 CALCOLO DEL FATTORIALE

```
[5]: # Chiede all'utente di inserire un numero intero
n = int(input("Inserisci un numero intero: "))

# Inizializza la variabile per il calcolo del fattoriale
fattoriale = 1

# Verifica se il numero è negativo
if n < 0:
    print("Il numero è negativo.")
# Se il numero è zero, il fattoriale è per convenzione 1
elif n == 0:
    print("Il fattoriale di zero è 1.")
else:
    # Calcola il fattoriale usando un ciclo for
    for numero in range(1, n + 1):
        fattoriale *= numero

# Stampa il risultato del fattoriale
print(f"Il fattoriale di {n} è {fattoriale}")
```

Inserisci un numero intero: 4
Il fattoriale di 4 è 24

2.16 CALCOLO DEL FATTORIALE CON GESTIONE DEI NUMERI NEGATIVI

```
[6]: # Chiede all'utente di inserire un numero intero
n = int(input("Inserisci un numero intero: "))

# Inizializza la variabile per il calcolo del fattoriale
fattoriale = 1

# Verifica se il numero è negativo
if n < 0:
    print("Il numero è negativo.")
# Se il numero è zero, il fattoriale è per convenzione 1
elif n == 0:
    print("Il fattoriale di zero è 1.")
else:
    # Calcola il fattoriale usando un ciclo for
    for numero in range(1, n + 1):
        fattoriale *= numero

# Stampa il risultato del fattoriale
print(f"Il fattoriale di {n} è {fattoriale}")
```

Inserisci un numero intero: -3
Il numero è negativo.

Il fattoriale di -3 è 1

2.17 CALCOLO DELLA SOMMA DEI NUMERI PARI FINO A “N”

```
[7]: # Chiedere all'utente di inserire un numero intero positivo N
N = int(input("Inserisci un numero intero positivo N:"))

#Inizializzare la somma a zero
somma = 0

#Calcolare la somma dei primi N numeri pari
for numero in range(2, 2 * N +1, 2):
    somma += numero

print (f"La somma dei primi {N} numeri pari è {somma}")
```

Inserisci un numero intero positivo N:7

La somma dei primi 7 numeri pari è 56

2.18 GENERAZIONE DI UNA LISTA DEI NUMERI PARI FINO A “N”

```
[8]: #Chiedere all'utente di inserire un numero intero positivo N
N = int(input("Inserisci un numero intero positivo N: "))
lista=[]

#Calcolare la somma dei primi N numeri pari
for numero in range(2, 2 * N +1, 2):
    lista.append(numero)

print(lista)
```

Inserisci un numero intero positivo N: 9

[2, 4, 6, 8, 10, 12, 14, 16, 18]

2.19 CONTEGGIO DELLE VOCI IN UNA FRASE O PAROLA

```
[9]: #Chiedi all'utente di inserire una frase o una parola
frase = input ("Inserisci una frase o una parola: ").lower() #Converti tutto in
#Lower =Serve per ridurre le stringhe di testo minuscole
#Inizializza il contatore delle vocali
conteggio_vocali = 0

#Definisci le vocali da cercare
vocali ="aeiou"

#Scansiona ogni carattere nella frase
for carattere in frase:
    #Verifica se il carattere è una vocale
    if carattere in vocali:
```

```

    conteggio_vocali += 1

#Stampa il conteggio delle vocali
print (f"Nella frase inserita ci sono {conteggio_vocali} vocali.")

```

Inserisci una frase o una parola: Redouane
 Nella frase inserita ci sono 5 vocali.

2.20 GIOCO DELL'INDOVINA IL NUMERO DEL DADO

```

[14]: import random
      # Genera un numero da 1 a 6 (simulando il lancio di un dado)
      numero_dado =random.randint(1, 6)

      #Chiedi all'utente di indovinare il numero
      indovina =int(input("Indovina il numero del dado (da 1 a 6): "))

      #Verifica se l'utente ha indovinato correttamente
      if indovina <1 or indovina >6:
          print("numero non ammesso")
      elif indovina ==numero_dado:
          print(f"Complimenti! Il numero del dado era {numero_dado}. Hai indovinato!")
      else:
          print(f"Mi dispiace, il numero del dado era {numero_dado}. Meglia fortuna,
            ↪alla prossima!")

```

Indovina il numero del dado (da 1 a 6): 5
 Mi dispiace, il numero del dado era 2. Meglia fortuna alla prossima!

2.21 ITERAZIONE DI SALUTO CON IL TUO NOME

```

[2]: import datetime

      today = datetime.datetime.today()
      print (f"oggi è il giorno: {today: %d %m %y} ore: {today: %H %M %S}")

```

oggi è il giorno: 10 11 23 ore: 09 57 07

2.22 SIMULAZIONE CRESCITA POPOLAZIONALE

```

[23]: #Inizializza la popolazione e gli anni
      popolazione=int(input("Inserire la popolazione iniziale: "))
      anni=int(input("Inserire il numero degli anni da simulare: "))
      #Tasso di natalità e di mortalità (percentuale annua)
      tasso_natalità=float(input("Inserisci il tasso di natalità: "))
      tasso_mortalità=float(input("Inserisci il tasso di mortalità: "))

      #Simulazione della crescita della popolazione
      for anno in range(anni):

```

```

nascite = int(popolazione*tasso_natalità)
morti = int(popolazione*tasso_mortalità)
popolazione += (nascite-morti)
print(f"Anno {anno + 1}: popolazione={int(popolazione)}") #se il print non è
↳dentro la dentazione da direttammete il risultato finale
print("Simulazione completata")

```

Inserire la popolazione iniziale: 20000
 Inserire il numero degli anni da simulare: 2
 Inserisci il tasso di natalità: 3
 Inserisci il tasso di mortalità: 2
 Anno 1: popolazione=40000
 Anno 2: popolazione=80000
 Simulazione completata

2.23 CONVERTITORE DI UNITÀ DI MISURA

```

[24]: #Convertitore di unita di Misura
print ("Benvenuto nel convertitore di Unità di Misura!")
scelta = input("Cosa desideri convertire? (metri/piedi/chilogrammi/libbre)").
↳lower()

if scelta == "metri":
    valore = float (input("Inserisci il valore in metri"))
    risultato = valore * 3.28084
    print(f"{valore} metri corrispondono a {risultato}piedi.")
elif scelta == "piedi":
    valore = float (input ("Inserisci il valore in piedi: "))
    risultati = valore / 3.28084
    print(f"{valore} piedi corrispondono a {risultato} metri.")
if scelta == "chilogrammi":
    valore = float (input("Inserisci il valore in chilogrammi"))
    risultato = valore * 2.20562
    print(f"{valore} metri corrispondono a {risultato} libbre.")
elif scelta == "libbre":
    valore = float (input ("Inserisci il valore in libbre: "))
    risultati = valore / 2.20562
    print(f"{valore} libbre corrispondono a {risultato} chilogrammi.")
else:
    print("Scelta non valida. Scegli tra 'metri', 'piedi', 'chilogrammi' o
↳'libbre'.")

```

Benvenuto nel convertitore di Unità di Misura!
 Cosa desideri convertire? (metri/piedi/chilogrammi/libbre)METRI
 Inserisci il valore in metri76
 76.0 metri corrispondono a 249.34384piedi.
 Scelta non valida. Scegli tra 'metri', 'piedi', 'chilogrammi' o 'libbre'.

2.24 CALCOLO DELL'ESIMO NUMERO DI FIBONACCI

```
[26]: #Fibonacci
n = int (input("Inserisci un numero n per calcolare l'esimo numero di Fibonacci_
↵"))
# Inizializzare le variabili per i primi due numeri di Fibonacci
a=0
b=1
c=1
# Calcolare l'n-esimo numero di Fibonacci
if n <= 0:
    print("Il numero deve essere maggiore di zero.")
elif n ==1:
    risultato = a
else:
    for iterazione in range(n-3):
        a = b
        b = c
        c = a + b
    risultato = c
# Stampare l'n-esimo numeri di Fibonacci
print("l'n-esimo numero di Fibonacci è: ", risultato)
```

Inserisci un numero n per calcolare l'esimo numero di Fibonacci 6
l'n-esimo numero di Fibonacci è: 5

3 FUNZIONI COSTUM

```
[7]: def Fibonacci(n):
    # Inizializza la serie di Fibonacci con i primi due termini
    fib_series = [0, 1]

    # Continua ad aggiungere nuovi termini finché la lunghezza della serie è_
    ↵inferiore a n
    while len(fib_series) < n:
        # Aggiunge il nuovo termine sommando gli ultimi due termini della serie
        fib_series.append(fib_series[-1] + fib_series[-2])

    # Restituisce la serie di Fibonacci fino all'n-esimo termine
    return fib_series
```

```
[8]: Fibonacci (10)
```

```
[8]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

3.1 CALCOLO DELLA SERIE DI FIBONACCI

```
[9]: # Chiede all'utente di inserire il numero di termini desiderati per la serie di
    ↪Fibonacci
n = int(input("Inserisci il numero di termini della serie di Fibonacci: "))

# Verifica se l'utente ha inserito un numero positivo
if n <= 0:
    print("Inserisci un numero positivo.")
else:
    # Chiama la funzione Fibonacci per generare la serie fino all'n-esimo termine
    result = Fibonacci(n)

    # Stampa la serie di Fibonacci
    print(result)
```

Inserisci il numero di termini della serie di Fibonacci: 6
[0, 1, 1, 2, 3, 5]

3.2 CALCOLATRICE DI AREE

```
[34]: import math #libreria math di python

def calcola_area_cerchio(raggio):
    return math.pi * (raggio ** 2) #math.pi = pi
def calcola_area Rettangolo(base, altezza):
    return base * altezza
def calcola_area_triangolo(base, altezza):
    return (base * altezza) / 2

[33]: # Stampa un messaggio di benvenuto
print("Benvenuto nella Calcolatrice di Aree!")

# Chiede all'utente di scegliere quale area calcolare
scelta = input("Vuoi calcolare l'area di un cerchio (c), rettangolo (r) o
    ↪triangolo (t)").lower()

# Gestisce la scelta dell'utente
if scelta == 'c':
    # Chiede all'utente di inserire il raggio del cerchio
    raggio = float(input("Inserisci il raggio del cerchio: "))
    # Chiama la funzione calcola_area_cerchio (che dovrebbe essere definita)
    area = calcola_area_cerchio(raggio)
    # Stampa il risultato
    print(f"L'area del cerchio è {area:.2f}")
elif scelta == 'r':
    # Chiede all'utente di inserire la base e l'altezza del rettangolo
    base = float(input("Inserisci la base del rettangolo: "))
```

```

altezza = float(input("Inserisci l'altezza del rettangolo: "))
# Chiama la funzione calcola_area_rettangolo (che dovrebbe essere definita)
area = calcola_area_rettangolo(base, altezza)
# Stampa il risultato
print(f"L'area del rettangolo è {area:.2f}")
elif scelta == 't':
    # Chiede all'utente di inserire la base e l'altezza del triangolo
    base = float(input("Inserisci la base del triangolo: "))
    altezza = float(input("Inserisci l'altezza del triangolo: "))
    # Chiama la funzione calcola_area_rettangolo (forse dovrebbe essere
    ↳calcola_area_triangolo?)
    area = calcola_area_rettangolo(base, altezza)
    # Stampa il risultato
    print(f"L'area del triangolo è {area:.2f}")
else:
    # Se l'utente ha inserito una scelta non valida
    print("Scelta non valida. Si prega di inserire 'c', 'r' o 't'.")

```

Benvenuto nella Calcolatrice di Aree!

Vuoi calcolare l'area di un cerchio (c), rettangolo (r) o triangolo (t)R

Inserisci la base del rettangolo: 9

Inserisci l'altezza del rettangolo: 12

L'area del rettangolo è 108.00

3.3 CALCOLATORE DI INTERESSI COMPOSTI

```

[35]: # Questa è una funzione che calcola l'importo finale di un investimento
# basato sull'importo iniziale, il tasso di interesse e il numero di periodi di
    ↳investimento.
def calcola_interessi(importo_iniziale, tasso_interesse, periodi_investimento):
    # Calcola l'importo finale utilizzando la formula dell'interesse composto.
    importo_finale = importo_iniziale * (1 + tasso_interesse / 100) **
    ↳periodi_investimento

    # Restituisce l'importo finale calcolato.
    return importo_finale

```

```

[37]: #Stampa un messaggio di benvenuto
print("Benvenuto nel Calcolatore Interessi")

#Definisci una funzione per calcolare l'importo finale con gli interessi
def calcola_interessi(importo, tasso, periodo):
    # Calcola l'importo finale con gli interessi usando la formula degli
    ↳interessi composti
    importo_finale = importo * (1 + tasso / 100) ** periodo
    return importo_finale

```

```

#Chiedi all'utente di inserire l'importo iniziale, il tasso di interesse e il
↳ periodo di investimento
importo = float(input("Inserisci l'importo iniziale: "))
tasso = float(input("Inserisci il tasso di interesse annuale (%): "))
periodo = int(input("Inserisci il periodo di investimento (anni): "))

#Chiama la funzione calcola_interessi per calcolare l'importo finale
importo_finale = calcola_interessi(importo, tasso, periodo)

#Stampa l'importo finale formattato con due decimali
print(f"L'importo finale dopo {periodo} anni è di {importo_finale:.2f} euro.")

```

Benvenuto nel Calcolatore Interessi
 Inserisci l'importo iniziale: 12000
 Inserisci il tasso di interesse annuale (%): 20
 Inserisci il periodo di investimento (anni): 2
 L'importo finale dopo 2 anni è di 17280.00 euro.

3.4 CALCOLO DELLA FORZA GRAVITAZIONALE

```

[39]: def calcola_forza_gravitazionale(massa1, massa2, distanza):
        # Definisce la costante gravitazionale
        G = 6.67430e-11 # Costante gravitazionale in N(m/kg)^2

        # Calcola la forza gravitazionale utilizzando la legge di gravitazione
        ↳ universale
        forza_gravitazionale = (G * massa1 * massa2) / (distanza**2)

        # Restituisce il risultato
        return forza_gravitazionale

[40]: #Esempio di utilizzo
        massa_terra=5.972e24 #kg
        massa_luna=7.342e22 #kg
        distanza_terra_luna=384400000 #metri

        forza= calcola_forza_gravitazionale(massa_terra, massa_luna, distanza_terra_luna)
        print(f"Forza gravitazionale tra la Terra e la Luna: {forza} Newton")

```

Forza gravitazionale tra la Terra e la Luna: 1.9804922390990566e+20 Newton

4 ANAGRAMMI

4.1 RISOLUTORE DI ANAGRAMMI

```
[41]: from itertools import permutations
```

```
k = 0
```

```
def trova_anagrammi(parola):  
    anagrammi = ["".join(p) for p in permutations(parola)]  
    return anagrammi
```

```
[45]: # Stampa un messaggio di benvenuto  
print("Benvenuto nel Risolutore di Anagrammi!")  
  
# Chiede all'utente di inserire una parola  
parola_input = input("Inserisci una parola: ").strip().lower()  
  
# Verifica se la parola ha almeno 2 caratteri  
if len(parola_input) < 2:  
    print("Inserisci una parola con almeno 2 caratteri.")  
else:  
    # Chiama la funzione trova_anagrammi (che dovrebbe essere definita)  
    anagrammi = trova_anagrammi(parola_input)  
  
    # Inizializza una variabile per contare gli anagrammi  
    k = 0  
  
    # Stampa gli anagrammi e conta quelli diversi dalla parola di input  
    for elemento in anagrammi:  
        if elemento != parola_input:  
            k += 1  
            print(elemento)  
  
    # Stampa il numero totale di anagrammi  
    print(f"Gli anagrammi di '{parola_input}' sono: {k}")
```

Benvenuto nel Risolutore di Anagrammi!

Inserisci una parola: Ciao

ciao

caio

caoi

coia

coai

icaio

icoa

iaco

iaoc


```
ioca
ioac
acio
acoi
aico
aioc
aoci
aoic
ocia
ocai
oica
oiac
oaci
oaic
Gli anagrammi di 'ciao' sono: 40386
```

4.2 CONTEGGIO DELLE LETTERE NELLA FRASE

```
[46]: # Chiedi all'utente di inserire una frase
frase = input("Inserisci una frase: ")

#Converti la frase in minuscoli per evitare problemi di maiuscole/minuscole
frase = frase.lower()

#Inizializza una lista di lettere dell'alfabeto
alfabeto = 'abcdefghijklmnopqrstuvwxyz'

#Inizializza un dizionario per tenere traccia del conteggio delle lettere
conteggio_lettere = {}

#Itera attraverso ciascuna lettera dell'alfabeto
for lettera in alfabeto:
    #Conta quante volte appare la lettera nella frase
    conteggio = frase.count(lettera)

    #Aggiungi la lettera e il conteggio al dizionario se la lettera appare
    if conteggio > 0:
        conteggio_lettere[lettera] = conteggio
#Stampa il conteggio delle lettere in un formato leggibile
for lettera, conteggio in conteggio_lettere.items():
    print(f"{lettera}: {conteggio} ")
```

```
Inserisci una frase: Ciao Marrocco
a: 2
c: 3
i: 1
m: 1
o: 3
```

r: 2

4.3 CONTEGGIO DEI PRODOTTI

```
[47]: conteggio lettere.items()
```

```
[47]: dict_items([('a', 2), ('c', 3), ('i', 1), ('m', 1), ('o', 3), ('r', 2)])
```

```
[48]: Prodotti={}
      Prodotti["pan bauletto"]=2
      Prodotti["coca cola"]=3
```

```
[49]: Prodotti
```

```
[49]: {'pan bauletto': 2, 'coca cola': 3}
```

4.4 CONVERTITORE DI VALUTE

```
[50]: # Definizione dei tassi di cambio
tassi_di_cambio = {
    "dollari": 1.0,
    "euro": 0.85,
    "yen": 110.41,
    # Aggiungi altre valute e tassi di cambio se necessario
}

# Chiedi all'utente di inserire l'importo, la valuta di partenza e la valuta di
↳ destinazione
try:
    importo = float(input("Inserisci l'importo da convertire: "))
    valuta_di_partenza = input("Inserisci la valuta di partenza: ").lower()
    valuta_destinazione = input("Inserisci la valuta di destinazione: ").lower()

    # Verifica se le valute sono nel dizionario dei tassi di cambio
    if valuta_di_partenza in tassi_di_cambio and valuta_destinazione in
↳ tassi_di_cambio:
        # Calcola il tasso di cambio e l'importo convertito
        tasso_di_cambio = tassi_di_cambio[valuta_destinazione] /
↳ tassi_di_cambio[valuta_di_partenza]
        importo_convertito = importo * tasso_di_cambio

        # Stampa il risultato
        print(f"{importo} {valuta_di_partenza} sono equivalenti a
↳ {importo_convertito:.2f} {valuta_destinazione}")
    else:
        print("Valute non supportate. Assicurati di inserire valute valide.")
except ValueError:
    print("Inserisci un importo valido.")
```

```
Inserisci l'importo da convertire: 1000
Inserisci la valuta di partenza: euro
Inserisci la valuta di destinazione: dollari
1000.0 euro sono equivalenti a 1176.47 dollari
```

4.5 OROLOGIO MONDIALE

```
[1]: from datetime import datetime
import pytz

# Stampa un messaggio di benvenuto
print("Benvenuto nell'Orologio mondiale!")

# Definisci le città e i relativi fusi orari
citta_fusi_orari = {
    "New York": "America/New_York",
```

```

    "Londra": "Europe/London",
    "Tokyo": "Asia/Tokyo",
    "Sydney": "Australia/Sydney",
    "Rio de Janeiro": "America/Sao_Paulo",
}

# Loop principale per consentire all'utente di effettuare scelte multiple
while True:
    # Stampa le città disponibili
    print("\nCittà disponibili: ")
    for città in città_fusi_orari.keys():
        print(città)

    # Chiede all'utente di inserire il nome della città o 'esci' per uscire
    scelta_città = input("Inserisci il nome della città per visualizzare l'ora,
    ↳(o 'esci' per uscire): ").strip()

    # Verifica se l'utente vuole uscire
    if scelta_città.lower() == 'esci':
        break

    # Verifica se la città inserita è valida
    if scelta_città in città_fusi_orari:
        # Ottiene il fuso orario corrispondente alla città
        fuso_orario = pytz.timezone(città_fusi_orari[scelta_città])

        # Ottiene l'ora corrente nella città scelta
        ora_corrente = datetime.now(fuso_orario)

        # Stampa l'ora corrente form

```

Benvenuto nell'Orologio mondiale!

Città disponibili:

New York

Londra

Tokyo

Sydney

Rio de Janeiro

Inserisci il nome della città per visualizzare l'ora (o 'esci' per uscire):

Tokyo

L'ora corrente a Tokyo è: 02:38:05

Città disponibili:

New York

Londra

Tokyo

Sydney

Rio de Janeiro
Inserisci il nome della città per visualizzare l'ora (o 'esci' per uscire):
Londra
L'ora corrente a Londra è: 17:38:22

Città disponibili:
New York
Londra
Tokyo
Sydney
Rio de Janeiro
Inserisci il nome della città per visualizzare l'ora (o 'esci' per uscire): esci

4.6 FUNZIONE PRINCIPALE PAOLO

```
[2]: #Funzione principale può avere qualsiasi nome
def paolo():
    print("Mi chiamo Paolo")

[3]: if __name__ == "__main__": #è una condizione logica che "si verifica sempre"

    paolo()
```

Mi chiamo Paolo

```
[4]: #Funzione principale può avere qualsiasi nome
def main():
    print(f"la funzione principale del codice è stata seguita, in questa_
    ↳funzione possono essere presenti precedentemente create")

if __name__ == "__main__":
    main()
```

la funzione principale del codice è stata seguita, in questa funzione possono essere presenti precedentemente create

4.7 PROGRAMMA DI VALUTAZIONE DEL BMI

```
[1]: #main
#Funzione per il calcolo del BMI

def calcola_bmi(peso, altezza):
    return peso / (altezza ** 2)

#Funzione per la valutazione del BMI

def valuta_bmi(bmi):
    if bmi < 18.5:
        return "Sottopeso"
```

```

elif 18.5 <= bmi < 24.9:
    return "Normapeso"
elif 25 <= bmi < 29.9:
    return "Sovrappeso"
else:
    return "Obeso"

```

```

[4]: # Importa le funzioni calcola_bmi e valuta_bmi (che dovrebbero essere definite)
from funzioni_bmi import calcola_bmi, valuta_bmi

# Funzione principale
def main():
    # Chiede all'utente di inserire il numero di persone da valutare
    n = int(input("Inserisci il numero di persone da valutare: "))

    # Ciclo for per iterare attraverso il numero di persone specificato
    for persone in range(n):
        print("Benvenuto nella Calcolatrice BMI!")

        # Chiede all'utente di inserire peso e altezza
        peso = float(input("Inserisci il tuo peso in chilogrammi: "))
        altezza = float(input("Inserisci la tua altezza in metri: "))

        # Chiama le funzioni per calcolare il BMI e valutare il BMI
        bmi = calcola_bmi(peso, altezza)
        valutazione = valuta_bmi(bmi)

        # Stampa il risultato
        print(f"Il tuo BMI è {bmi}")
        print(f"Valutazione: {valutazione}")

    # Verifica se lo script è in esecuzione come programma principale
    if __name__ == "__main__":
        main()

```

```

Inserisci numero di persone da valutare: 1
Benvenuto nella Calcolatrice BMI!
Inserisci il tuo peso in chilogrammi: 65
Inserisci la tua altezza in metri: 1.85
Il tuo Bmi è 18.991964937910883
Valutazione: Normapeso

```

4.8 CONVERTITORE DI UNITÀ DI MISURA

```

[6]: # Funzione per la conversione da metri a piedi
def metri_a_piedi(metri):
    return metri * 3.28084

```

```

# Funzione per la conversione da piedi a metri
def piedi_a_metri(piedi):
    return piedi / 3.28084

# Funzione per la conversione da chilogrammi a libbre
def chilogrammi_a_libbre(chilogrammi):
    return chilogrammi * 2.20462

# Funzione per la conversione da libbre a chilogrammi
def libbre_a_chilogrammi(libbre):
    return libbre / 2.20462

# Funzione di selezione per gestire le conversioni in base all'unità di misura
↳ scelta
def selezione(scelta):
    if scelta == "metri":
        valore = float(input("Inserisci il valore in metri: "))
        risultato = metri_a_piedi(valore)
        print(f"{valore:.3f} metri corrispondono a {risultato:.3f} piedi.")
    elif scelta == "piedi":
        valore = float(input("Inserisci il valore in piedi: "))
        risultato = piedi_a_metri(valore)
        print(f"{valore:.3f} piedi corrispondono a {risultato:.3f} metri.")
    elif scelta == "chilogrammi":
        valore = float(input("Inserisci il valore in chilogrammi: "))
        risultato = chilogrammi_a_libbre(valore)
        print(f"{valore:.3f} chilogrammi corrispondono a {risultato:.3f} libbre.
↳ ")
    elif scelta == "libbre":
        valore = float(input("Inserisci il valore in libbre: "))
        risultato = libbre_a_chilogrammi(valore)
        print(f"{valore:.3f} libbre corrispondono a {risultato:.3f} chilogrammi.
↳ ")
    else:
        print("Scelta non valida. Scegli tra 'metri', 'piedi', 'chilogrammi' o
↳ 'libbre'.")

# Chiede all'utente l'unità di misura per la conversione
unita_misura = input("Vuoi convertire da metri a piedi, da piedi a metri, da
↳ chilogrammi a libbre o da libbre a chilogrammi? ").strip().lower()

# Chiama la funzione di selezione con l'unità di misura scelta
selezione(unita_misura)

```

Vuoi convertire da metri a piedi, da piedi a metri, da chilogrammi a libbre o da libbre a chilogrammi? piedi
Inserisci il valore in piedi: 54

54.000 piedi corrispondono a 16.459 metri.

4.9 GESTIONE DELLE CALORIE CONSUMATE

```
[1]: # Dizionario con le calorie per 100 grammi di cibo
cibo_calorie = {
    "banana": 89,
    "mela": 52,
    "arancia": 43,
    # Altri cibi...
}

# Funzione per calcolare le calorie consumate
def calorie_consumate(cibo, quantita):
    if cibo not in cibo_calorie:
        print("Cibo non presente")
        return 0 # Ritorna 0 calorie se il cibo non è nel dizionario
    calorie_per_100g = cibo_calorie[cibo]
    calorie_totali = (calorie_per_100g / 100) * quantita
    return calorie_totali

# Funzione principale
def main():
    cibo_consumato = []

    while True:
        print("Menu")
        print("\n1. Aggiungi cibo consumato")
        print("2. Calcola calorie totali")
        print("3. Esci")

        scelta = input("Scegli un'opzione: ")

        if scelta == "1":
            print("\nCibi disponibili:")
            for cibo in cibo_calorie:
                print(cibo)
            cibo = input("Inserisci il cibo consumato: ").lower()
            quantita = float(input("Inserisci la quantita (in grammi): "))
            cibo_consumato.append((cibo, quantita))

        elif scelta == "2":
            calorie_totali = sum(calorie_consumate(c, q) for c, q in
→cibo_consumato)
            print(f"\nCalorie totali consumate: {calorie_totali} calorie")

        elif scelta == "3":
```



```

        break

    else:
        print("\nScelta non valida. Riprova.")

if __name__ == "__main__":
    main()

```

Menu

1. Aggiungi cibo consumato
2. Calcola calorie totali
3. Esci

Scegli un'opzione: 1

Cibi disponibili:

banana

mela

arancia

Inserisci il cibo consumato: banana

Inserisci la quantita (in grammi): 54

Menu

1. Aggiungi cibo consumato
2. Calcola calorie totali
3. Esci

Scegli un'opzione: 3

```

[2]: acquisti={}
    acquisti ["pan bauletto"]= 10
    acquisti ["nutella"]= 10

```

```

[3]: acquistidue= {
    "pan bauletto":10,
    "nutella":10,
}

```

5 GENERATORE DI PERSONAGGI

5.1 GENERATORE DI PERSONAGGI FANTASY

```

[4]: import random

# Liste di specie, classi, armi e abilità
speci = ["Elfo", "Umano", "Nano", "Orco", "Gnomo"]
classi = ["Guerriero", "Mago", "Ranger", "Ladro", "Chierico"]
armi = ["Spada", "Arco", "Bacchetta magica", "Ascia", "Daga"]

```

```

abilita = ["Furtività", "Magia dell'acqua", "Camuffamento", "Estrazione_
↳ mineraria", "Incantesimi di guarigione"]

# Genera un personaggio casuale
specie = random.choice(speci)
classe = random.choice(classi)
arma = random.choice(armi)
abilita_scelte = random.sample(abilita, random.randint(1, 3))

# Stampa il personaggio generato
print("Personaggio Fantasy Generato:")
print(f"Specie: {specie}")
print(f"Classe: {classe}")
print(f"Arma: {arma}")
print(f"Abilità: {' '.join(abilita_scelte)}")

```

Personaggio Fantasy Generato:
 Specie: Umano
 Classe: Chierico
 Arma: Bacchetta magica
 Abilità: Magia dell'acqua, Estrazione mineraria

5.2 GENERATORE DI PERSONAGGI

```

[6]: import random

# Estendi queste liste con attributi, tratti di personalità, sfondi e motivazioni
physical_traits = ["alto", "basso", "magro", "corpulento", "capelli lunghi",
↳ "capelli corti"]
personality_traits = ["gentile", "introverso", "estroverso", "avventuroso",
↳ "analitico"]
backgrounds = ["contadino", "nobile", "commerciante", "ladro", "mago"]
motivations = ["vendicare una perdita", "trovare l'amore", "arricchirsi",
↳ "scoprire la verità"]

def genera_personaggio():
    # Chiede all'utente di inserire il nome del personaggio
    nome = input("Inserisci il nome del personaggio: ")

    # Sceglie casualmente un aspetto fisico, un tratto di personalità, uno
    ↳ sfondo e una motivazione
    aspetto_fisico = random.choice(physical_traits)
    aspetto_personale = random.choice(personality_traits)
    sfondo = random.choice(backgrounds)
    motivazione = random.choice(motivations)

    # Crea una descrizione del personaggio

```

```
descrizione = f"{nome} è un personaggio {aspetto_fisico},  
↳{aspetto_personale}, di sfondo {sfondo} e con la motivazione di {motivazione}."  
  
# Stampa la descrizione del personaggio  
print(descrizione)  
  
# Chiama la funzione per generare un personaggio  
genera_personaggio()
```

Inserisci il nome del personaggio: Miao

Miao è un personaggio magro, gentile, di sfondo nobile e con la motivazione di scoprire la verità.