

# Esercizi (while)

## Esercizio 1

### Calcolare la media A

Chiedere all'utente di inserire dei numeri. Continuare a chiedere all'utente di inserire un numero sino a che non inserisce un numero negativo per fermare il ciclo. Dopo che l'utente ha inserito un numero negativo, quindi fermato il ciclo delle richeiste, stampare la media dei numeri inseriti.

```
In [30]: sum = 0
count = 0
number = float(input("Enter a number (enter negative number to stop): "))

while number >= 0:
    sum += number
    count += 1
    number = float(input("Enter a number (enter negative number to stop): "))

if count > 0:
    average = sum / count
    print(f"The average is: {average}")

The average is: 6.2
```

### Calcolare la media B

Fare in modo che ad ogni nuovo numero inserito, venga calcolata immediatamente la media tra tutti i numeri inseriti in precedenza e il nuovo.

```
In [36]: number = float(input("Enter a number (enter negative number to stop): "))
sum = number
count = 1

while number >= 0:
    number = float(input("Enter a number (enter negative number to stop): "))
    sum += number
    count += 1

    if number < 0:
        print("STOP")
    elif count > 0:
        average = sum / count
        print(f"The average is: {average}")

The average is: 10.0
The average is: 8.333333333333334
The average is: 8.75
The average is: 8.8
The average is: 9.0
The average is: 8.428571428571429
The average is: 8.625
The average is: 8.777777777777779
The average is: 8.9
STOP
```

## Esercizio 2

### Solo consonanti

Chiedere all'utente di inserire una stringa. Stampare solo le consonanti della stringa usando un loop while.

```
In [4]: s = input("Inserisci una stringa: ")
i = 0

while i < len(s):
    if s[i] not in "aeiouAEIOU":
        print(s[i], end="")
    i += 1

c cm stt l
```

## Esercizio 3

### Calcolare il fattoriale

Chiedere all'utente di inserire un numero intero positivo.

```
In [1]: n = int(input("Enter a positive number: "))
factorial = 1
i = 1

while i <= n:
    factorial *= i
    i += 1

print(f"The factorial of {n} is {factorial}")

The factorial of 6 is 720
```

## Esercizio 4

### Correzione operazioni

Chiedere all'utente di inserire due numeri interi.

Chiedere all'utente quale operazione eseguire sui due numeri, controllare che scelga un'operazione valida altrimenti richiedere quell'operazione in loop sino a che non inserisce un valore corretto.

Chiedere all'utente la risposta all'operazione scelta con i due numeri inseriti:

- Se la risposta è giusta terminare il programma
- Se è sbagliata chiedere la risposta in loop sino a che non è corretta

```
In [94]: print("Benvenuto!")

n1 = int(input("Inserire un numero intero: "))
n2 = int(input("Inserire un nuovo numero intero: "))

print("Su cosa desideri esercitarti?")
while True:
    operazione = input("a - addizioni, b - sottrazioni, c - moltiplicazioni, d - divisioni\n")
    if operazione == 'a' or operazione == 'b' or operazione == 'c' or operazione == 'd':
        break
    else:
        print("Scegliere una operazione valida\n")

while True:
    risposta = int(input("Inserire la risposta: "))
    if operazione == 'a':
        risultato = n1+n2
        if risposta == risultato:
            print("Bravissima/o!")
            break
        else:
            print("Risposta errata, riprova\n")
    elif operazione == 'b':
        risultato = n1-n2
        if risposta == risultato:
            print("Bravissima/o!")
            break
        else:
            print("Risposta errata, riprova\n")
    elif operazione == 'c':
        risultato = n1*n2
        if risposta == risultato:
            print("Bravissima/o!")
            break
        else:
            print("Risposta errata, riprova\n")
    elif operazione == 'd':
        risultato = int(n1/n2)
        if risposta == risultato:
            print("Bravissima/o!")
            break
        else:
            print("Risposta errata, riprova\n")
    else:
        print("Inserire una risposta valida!")

Benvenuto!
Su cosa desideri esercitarti?
Risposta errata, riprova

Bravissima/o!
```

## Esercizio 5

### Creazione di una lista di promozioni

Hai tre liste che rappresentano:

- studenti - I nomi degli studenti di una classe.
- voti - I voti corrispondenti a ciascun studente.
- promossi - Una lista vuota che conterrà i nomi degli studenti promossi.

Scrivi un programma che:

- Utilizzi un ciclo while per scorrere la lista degli studenti e dei loro voti.
- Controlli i voti degli studenti:
  - Se un voto è maggiore o uguale a 6, aggiungi lo studente alla lista promossi.
  - Altrimenti, stampa un messaggio indicando che lo studente non è stato promosso.
- Al termine, stampa la lista degli studenti promossi e di quelli non promossi.

Dati iniziali:

```
studenti = ["Alice", "Bob", "Charlie", "Diana", "Eve"]
voti = [7, 5, 6, 8, 4]
promossi = []
```

Esempio di output atteso:

```
Alice è stato/a promosso/a!

Bob non è stato/a promosso/a. Voto: 5

Charlie è stato/a promosso/a!

Diana è stato/a promosso/a!

Eve non è stato/a promosso/a. Voto: 4
```

Risultato finale:

```
Studenti promossi: ['Alice', 'Charlie', 'Diana']

Studenti non promossi: ['Bob', 'Eve']
```

```
In [61]: # Dati iniziali
studenti = ["Alice", "Bob", "Charlie", "Diana", "Eve"]
voti = [7, 5, 6, 8, 4]
promossi = []
non_promossi = []

# Inizializzazione dell'indice per il ciclo while
i = 0

while i < len(studenti):
    # Ottieni lo studente e il voto corrente
    studente = studenti[i]
    voto = voti[i]

    if voto >= 6:
        # Aggiungi lo studente ai promossi
        promossi.append(studente)
        print(f"{studente} è stato/a promosso/a! Voto: {voto}")
    else:
        # Aggiungi lo studente ai non promossi
        non_promossi.append(studente)
        print(f"{studente} non è stato/a promosso/a. Voto: {voto}")

    i += 1

# Risultato finale
print(f"Studenti promossi: {promossi}")
print(f"Studenti non promossi: {non_promossi}")

Alice è stato/a promosso/a! Voto: 7
Bob non è stato/a promosso/a. Voto: 5
Charlie è stato/a promosso/a! Voto: 6
Diana è stato/a promosso/a! Voto: 8
Eve non è stato/a promosso/a. Voto: 4
Studenti promossi: ['Alice', 'Charlie', 'Diana']
Studenti non promossi: ['Bob', 'Eve']
```

## Esercizio 6

### Gestione degli ordini in un magazzino

Hai tre liste che rappresentano:

- prodotti - I prodotti disponibili in magazzino.
- quantità - La quantità disponibile per ciascun prodotto.
- ordini - Gli ordini effettuati dai clienti (ogni ordine è un prodotto richiesto).

Scrivi un programma che:

- Scorra la lista degli ordini utilizzando un ciclo while.
- Controlli se il prodotto richiesto è disponibile nella lista prodotti e se la quantità è sufficiente.
  - Se il prodotto è disponibile e c'è abbastanza quantità, riduci il valore corrispondente nella lista quantità e stampa un messaggio di conferma.
  - Se il prodotto è disponibile ma non c'è abbastanza quantità, stampa un messaggio di scorta insufficiente.
  - Se il prodotto non è disponibile, stampa un messaggio che indichi che il prodotto non esiste.
- Al termine, stampa la lista aggiornata di quantità.

Dati iniziali:

```
prodotti = ["mela", "banana", "arancia", "pera"]
quantità = [10, 5, 8, 3]
ordini = ["mela", "arancia", "banana", "mela", "pera", "kiwi"]
```

Esempio di output atteso:

```
Ordine: mela - Confermato! Quantità rimanente: 9

Ordine: arancia - Confermato! Quantità rimanente: 7

Ordine: banana - Confermato! Quantità rimanente: 4

Ordine: mela - Confermato! Quantità rimanente: 8

Ordine: pera - Confermato! Quantità rimanente: 2

Ordine: kiwi - Prodotto non disponibile!

Quantità finale: [8, 4, 7, 2]
```

```
In [ ]: # Dati iniziali
prodotti = ["mela", "banana", "arancia", "pera"]
quantità = [10, 5, 8, 3]
ordini = ["mela", "arancia", "banana", "mela", "pera", "kiwi"]

# Inizializzazione dell'indice per il ciclo while
i = 0

while i < len(ordini):
    ordine = ordini[i]

    if ordine in prodotti:
        # ottieni l'indice del prodotto nella lista prodotti
        k = 0
        indice = -1
        while k < len(prodotti) and indice == -1:
            if prodotti[k] == ordine:
                indice = k
            k += 1
        # in alternativa si può anche usare indice = prodotti.index(ordine)

        if quantità[indice] > 0:
            # Aggiorna la quantità e stampa la conferma
            quantità[indice] -= 1
            print(f"Ordine: {ordine} - Confermato! Quantità rimanente: {quantità[indice]}")
        else:
            # Quantità insufficiente
            print(f"Ordine: {ordine} - Scorta insufficiente!")
    else:
        # Prodotto non trovato
        print(f"Ordine: {ordine} - Prodotto non disponibile!")

    i += 1

# Stampa finale delle quantità aggiornate
print(f"Quantità finale: {quantità}")
```