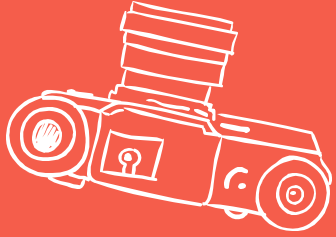
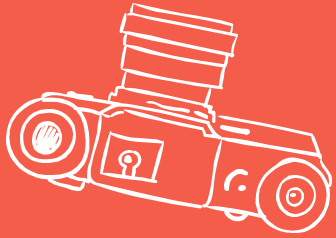


LABORATORIO DI PENSIERO COMPUTAZIONALE

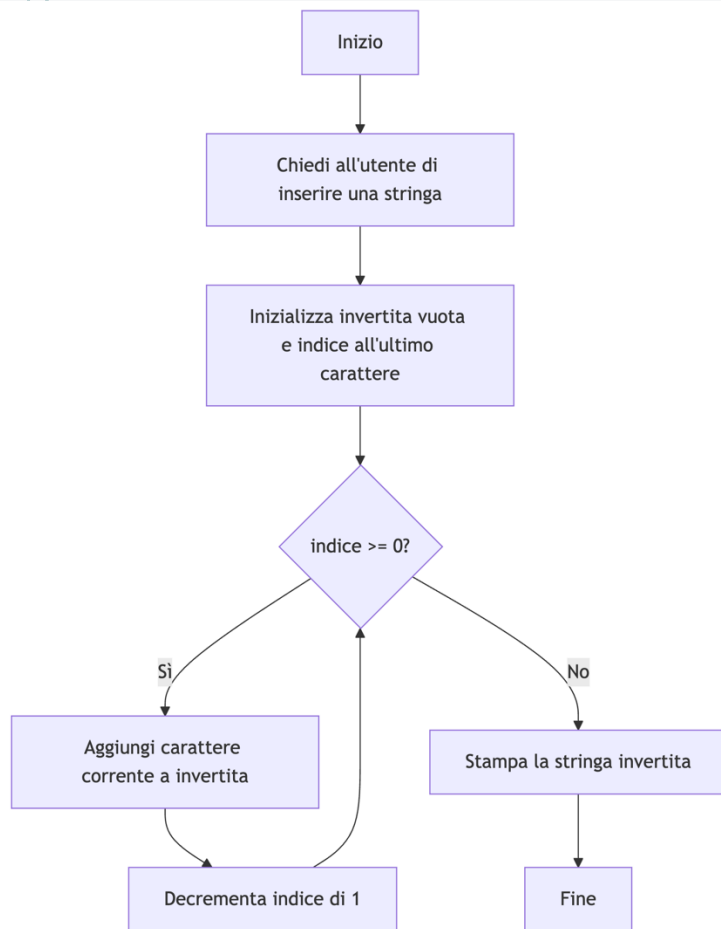


Simone Zennaro – simone.zennaro@unipd.it

STRINGHE



INVERTIRE UNA STRINGA: SCRIVI UN PROGRAMMA CHE PRENDA UNA STRINGA DALL'UTENTE E STAMPI I CARATTERI IN ORDINE INVERSO.



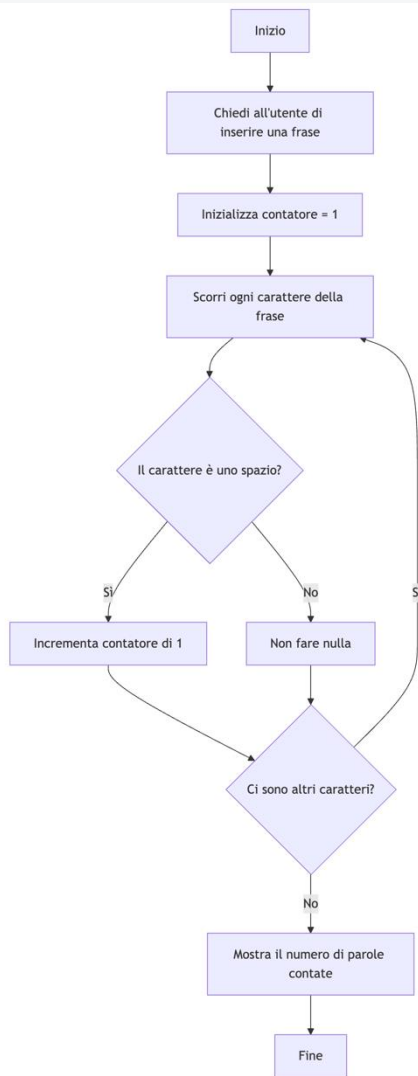
```
# Chiedi all'utente di inserire una stringa
testo = input("Inserisci una stringa: ")

# Inizializza variabili
invertita = ""
indice = len(testo) - 1 # partiamo dall'ultimo carattere

# Ciclo while per scorrere la stringa all'indietro
while indice >= 0:
    invertita += testo[indice]
    indice -= 1 # decrementa l'indice

# Mostra la stringa invertita
print("Stringa invertita:", invertita)
```

CONTARE LE PAROLE: SCRIVI UN PROGRAMMA CHE PRENDA UNA FRASE DALL'UTENTE E CONTI QUANTE PAROLE CI SONO NELLA FRASE. PUOI ASSUMERE CHE LE PAROLE SIANO SEPARATE DA SPAZI.



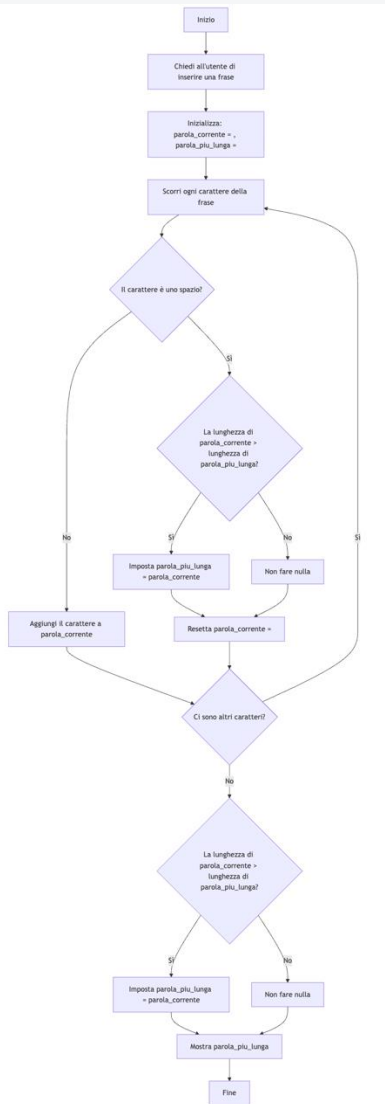
```
# Chiedi all'utente di inserire una frase
frase = input("Inserisci una frase: ")

# Inizializza contatore
contatore = 1 if frase else 0 # se la frase è vuota, contatore = 0

# Scorri ogni carattere della frase
for c in frase:
    if c == " ":
        contatore += 1

# Mostra il numero di parole
print("Numero di parole:", contatore)
```

TROVA LA PAROLA PIÙ LUNGA: SCRIVI UN PROGRAMMA CHE PRENDA UNA FRASE E TROVI LA PAROLA PIÙ LUNGA NELLA FRASE.



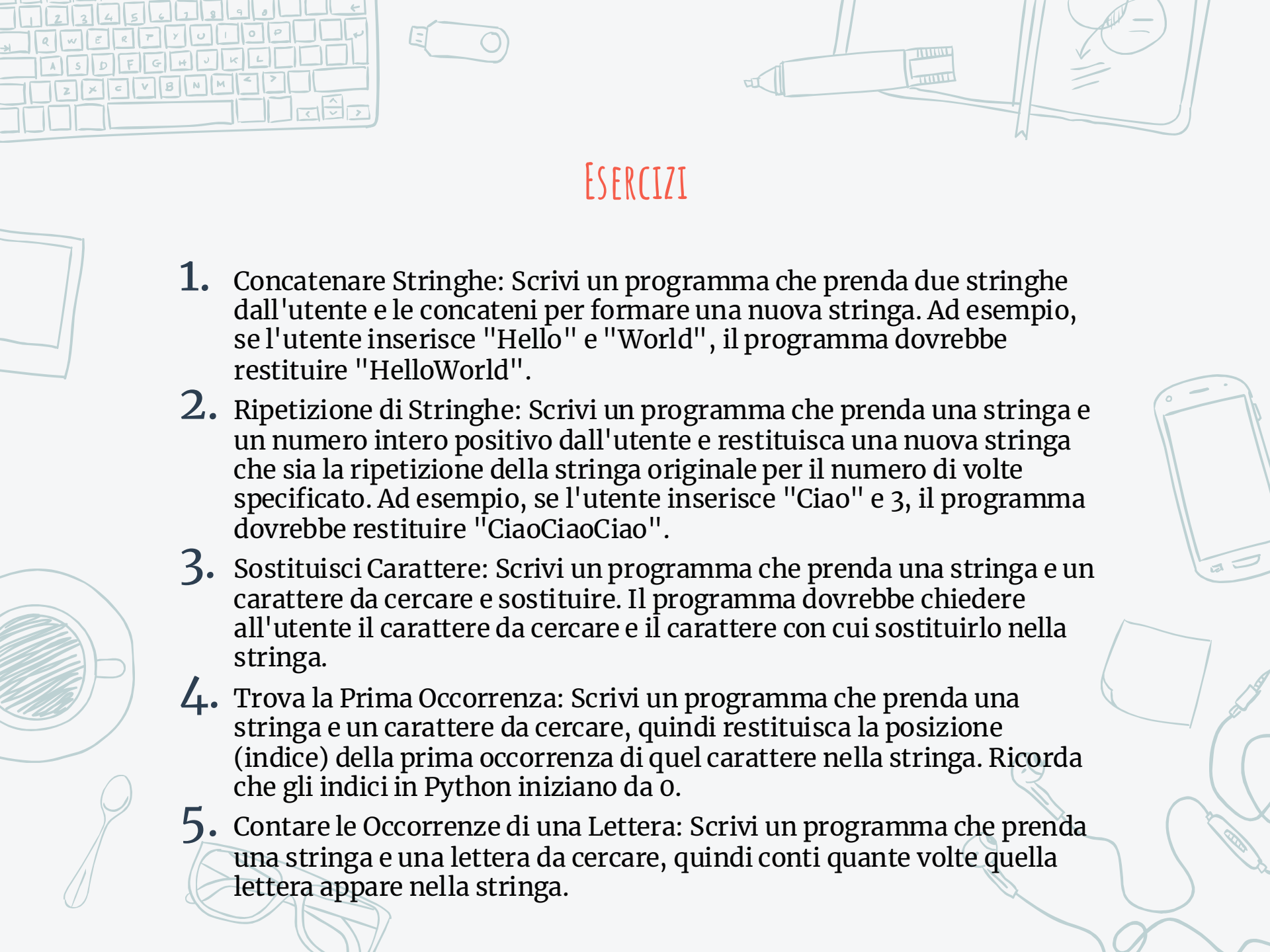
```
# Chiedi all'utente di inserire una frase
frase = input("Inserisci una frase: ")

# Inizializza variabili
parola_corrente = ""
parola_piu_lunga = ""

# Scorri ogni carattere della frase
for carattere in frase:
    if carattere != " ":
        # Aggiungi il carattere a parola_corrente
        parola_corrente += carattere
    else:
        # Controlla se parola_corrente è più lunga di parola_piu_lunga
        if len(parola_corrente) > len(parola_piu_lunga):
            parola_piu_lunga = parola_corrente
        # Resetta parola_corrente
        parola_corrente = ""

# Controllo finale per l'ultima parola (se la frase non termina con uno spazio)
if len(parola_corrente) > len(parola_piu_lunga):
    parola_piu_lunga = parola_corrente

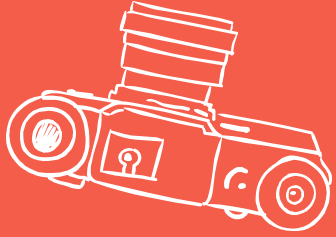
# Mostra la parola più lunga
print("La parola più lunga è:", parola_piu_lunga)
```



ESERCIZI

1. **Concatenare Stringhe:** Scrivi un programma che prenda due stringhe dall'utente e le concateni per formare una nuova stringa. Ad esempio, se l'utente inserisce "Hello" e "World", il programma dovrebbe restituire "HelloWorld".
2. **Ripetizione di Stringhe:** Scrivi un programma che prenda una stringa e un numero intero positivo dall'utente e restituisca una nuova stringa che sia la ripetizione della stringa originale per il numero di volte specificato. Ad esempio, se l'utente inserisce "Ciao" e 3, il programma dovrebbe restituire "CiaoCiaoCiao".
3. **Sostituisci Carattere:** Scrivi un programma che prenda una stringa e un carattere da cercare e sostituire. Il programma dovrebbe chiedere all'utente il carattere da cercare e il carattere con cui sostituirlo nella stringa.
4. **Trova la Prima Occorrenza:** Scrivi un programma che prenda una stringa e un carattere da cercare, quindi restituisca la posizione (indice) della prima occorrenza di quel carattere nella stringa. Ricorda che gli indici in Python iniziano da 0.
5. **Contare le Occorrenze di una Lettera:** Scrivi un programma che prenda una stringa e una lettera da cercare, quindi conti quante volte quella lettera appare nella stringa.

LISTE



LE LISTE IN PYTHON

```
ages = [19, 26, 23]
print(ages)

# Output: [19, 26, 23]
```

Le stringhe sono una sequenza di elementi

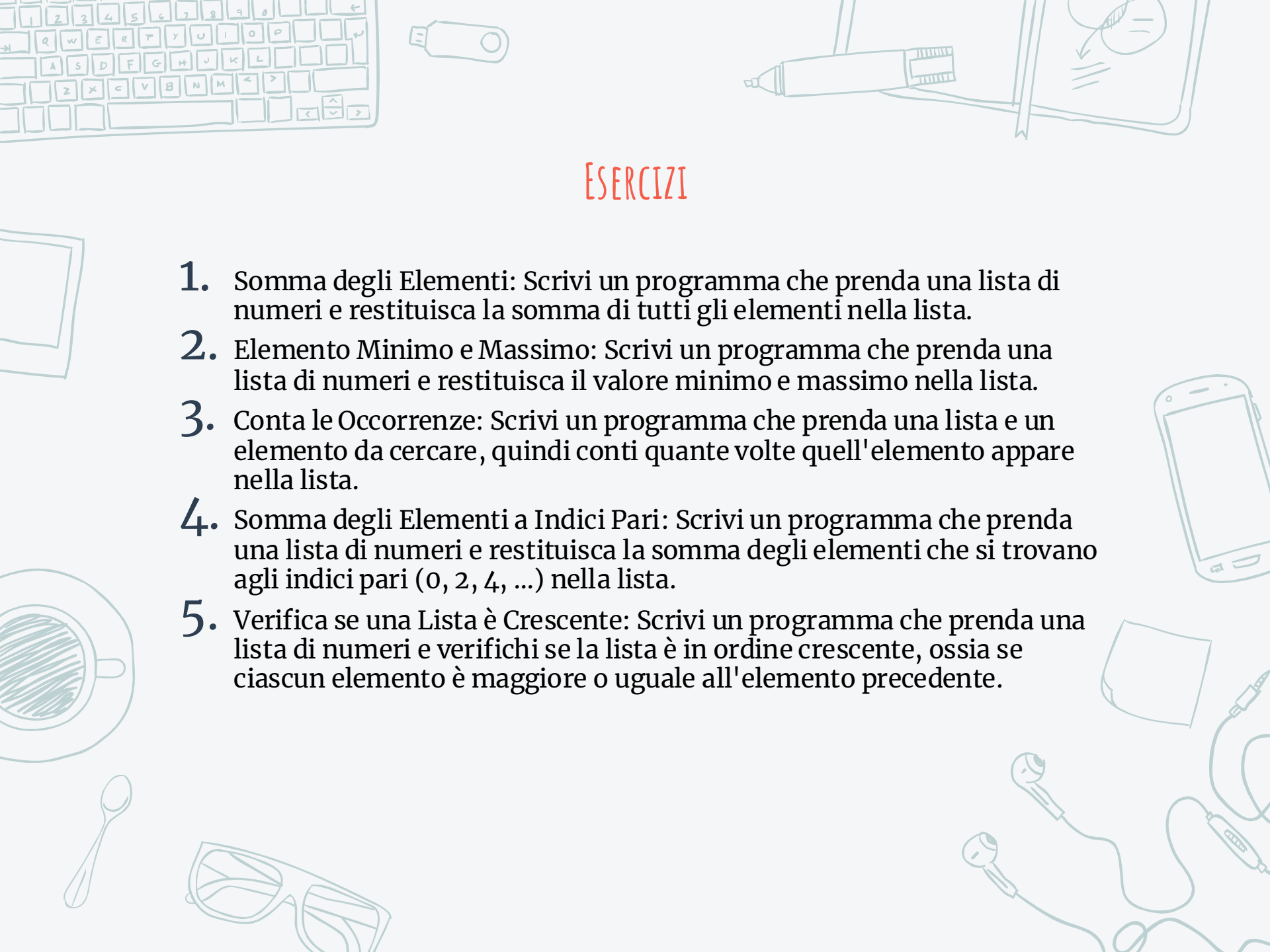
Possiamo accedere ai singoli elementi utilizzando un indice numerico che parte da 0

```
languages = ["Python", "Swift", "C++"]

# access item at index 0
print(languages[0]) # Python

# access item at index 2
print(languages[2]) # C++
```

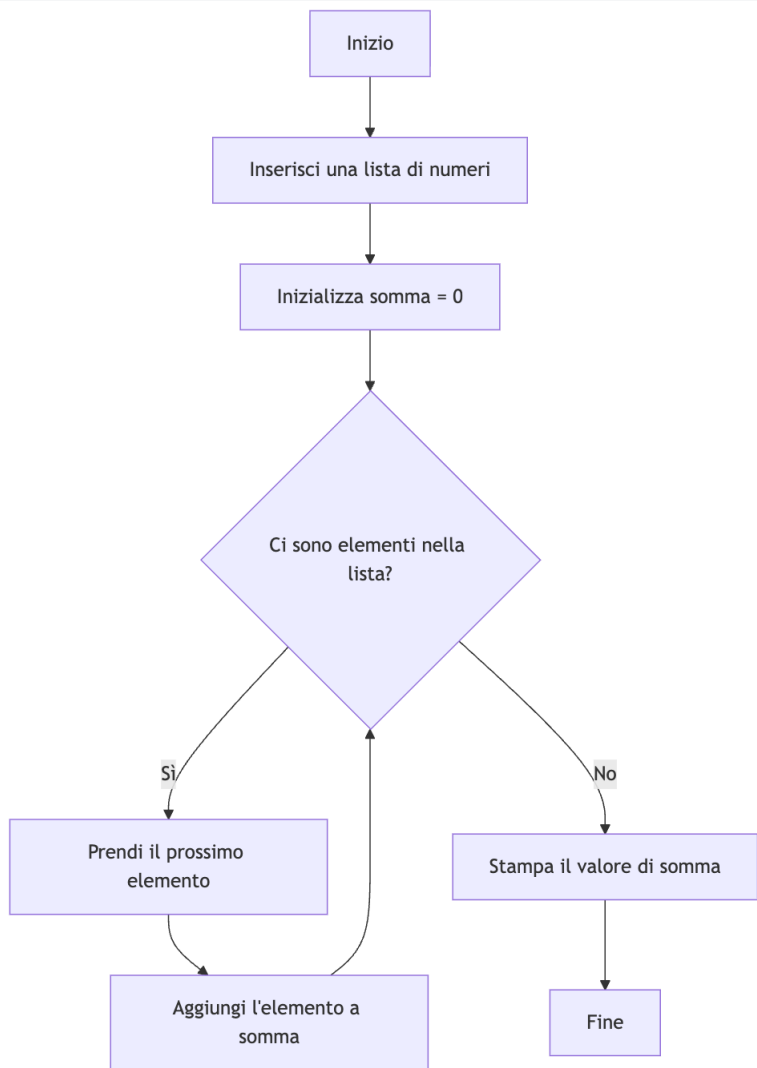
Le stringhe si comportano quindi come liste di caratteri



ESERCIZI

1. **Somma degli Elementi:** Scrivi un programma che prenda una lista di numeri e restituisca la somma di tutti gli elementi nella lista.
2. **Elemento Minimo e Massimo:** Scrivi un programma che prenda una lista di numeri e restituisca il valore minimo e massimo nella lista.
3. **Conta le Occorrenze:** Scrivi un programma che prenda una lista e un elemento da cercare, quindi conti quante volte quell'elemento appare nella lista.
4. **Somma degli Elementi a Indici Pari:** Scrivi un programma che prenda una lista di numeri e restituisca la somma degli elementi che si trovano agli indici pari (0, 2, 4, ...) nella lista.
5. **Verifica se una Lista è Crescente:** Scrivi un programma che prenda una lista di numeri e verifichi se la lista è in ordine crescente, ossia se ciascun elemento è maggiore o uguale all'elemento precedente.

SOMMA DEGLI ELEMENTI: SCRIVI UN PROGRAMMA CHE PRENDA UNA LISTA DI NUMERI E RESTITUISCA LA SOMMA DI TUTTI GLI ELEMENTI NELLA LISTA.



```
# Inizio
```

```
# Inserisci una lista di numeri  
numeri = [1, 2, 3, 4, 5]
```

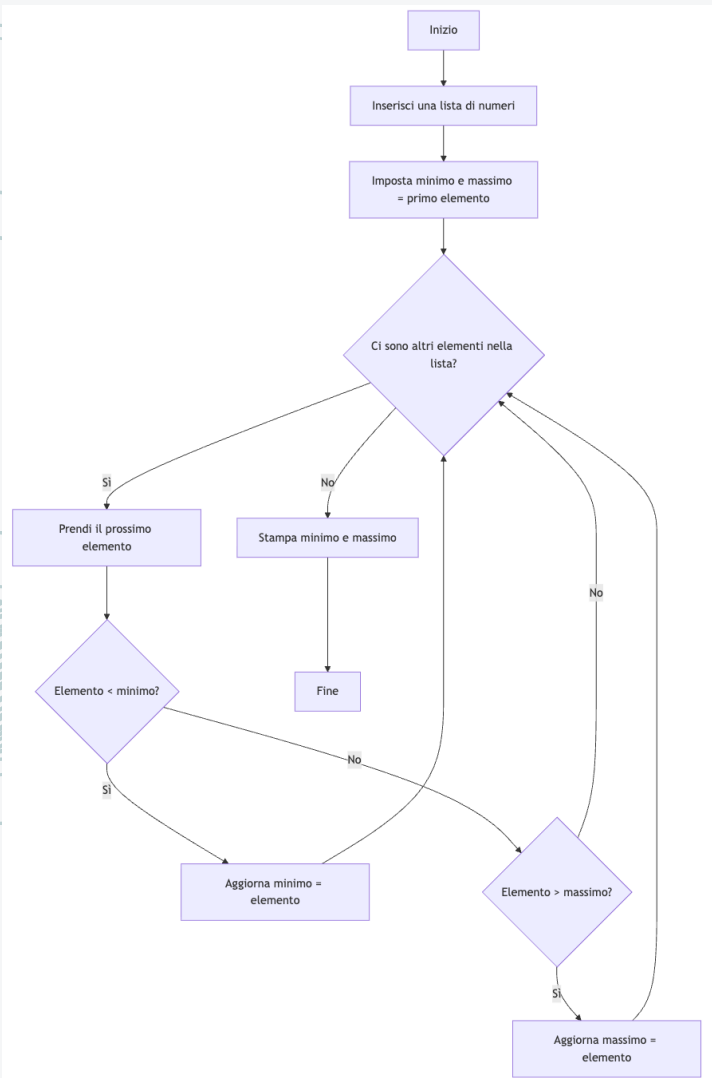
```
# Inizializza somma = 0  
somma = 0
```

```
# Ciclo sugli elementi della lista  
for elemento in numeri:  
    # Aggiungi l'elemento a somma  
    somma += elemento
```

```
# Stampa il valore di somma  
print("La somma degli elementi è:", somma)
```

```
# Fine
```

SOMMA DEGLI ELEMENTI: SCRIVI UN PROGRAMMA CHE PRENDA UNA LISTA DI NUMERI E RESTITUISCA LA SOMMA DI TUTTI GLI ELEMENTI NELLA LISTA.



```
# Inizio
```

```
# Inserisci una lista di numeri
numeri = [10, 3, 25, 7, 15, -2]
```

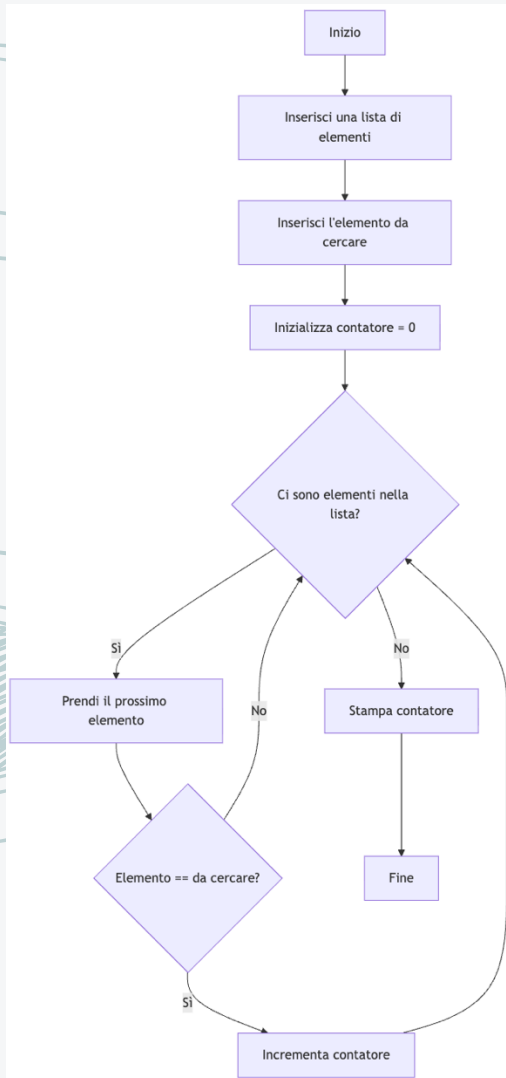
```
# Imposta minimo e massimo = primo elemento
minimo = numeri[0]
massimo = numeri[0]
```

```
# Ciclo sugli altri elementi della lista
for elemento in numeri[1:]:
    if elemento < minimo:
        minimo = elemento
    elif elemento > massimo:
        massimo = elemento
```

```
# Stampa i risultati
print("Il valore minimo è:", minimo)
print("Il valore massimo è:", massimo)
```

```
# Fine
```

CONTA LE OCCORRENZE: SCRIVI UN PROGRAMMA CHE PRENDA UNA LISTA
E UN ELEMENTO DA CERCARE, QUINDI CONTI QUANTE VOLTE
QUELL'ELEMENTO APPARE NELLA LISTA.



```
# Inizio

# Inserisci una lista di elementi
lista = [1, 2, 3, 2, 4, 2, 5]

# Inserisci l'elemento da cercare
da_cercare = 2

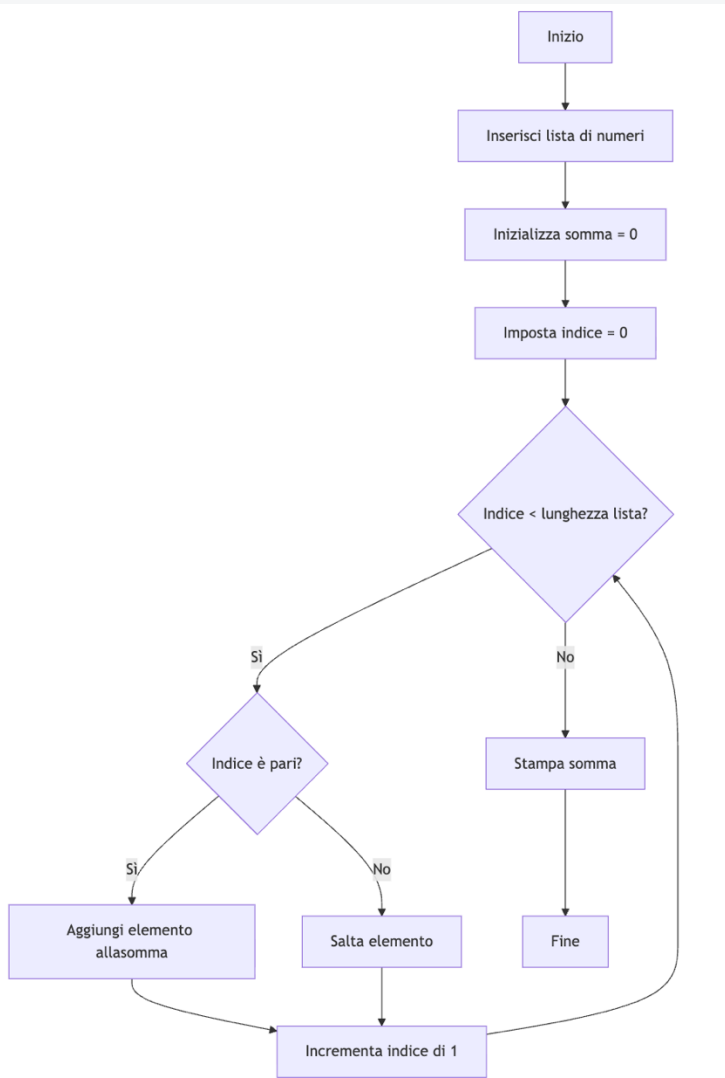
# Inizializza il contatore
contatore = 0

# Ciclo sugli elementi della lista
for elemento in lista:
    if elemento == da_cercare:
        contatore += 1

# Stampa il risultato
print(f"L'elemento {da_cercare} appare {contatore} volte nella lista.")

# Fine
```

SOMMA DEGLI ELEMENTI A INDICI PARI: SCRIVI UN PROGRAMMA CHE PRENDA UNA LISTA DI NUMERI E RESTITUISCA LA SOMMA DEGLI ELEMENTI CHE SI TROVANO AGLI INDICI PARI (0, 2, 4, ...) NELLA LISTA.



```
# Inizio

# Inserisci una lista di numeri
numeri = [10, 20, 30, 40, 50, 60]

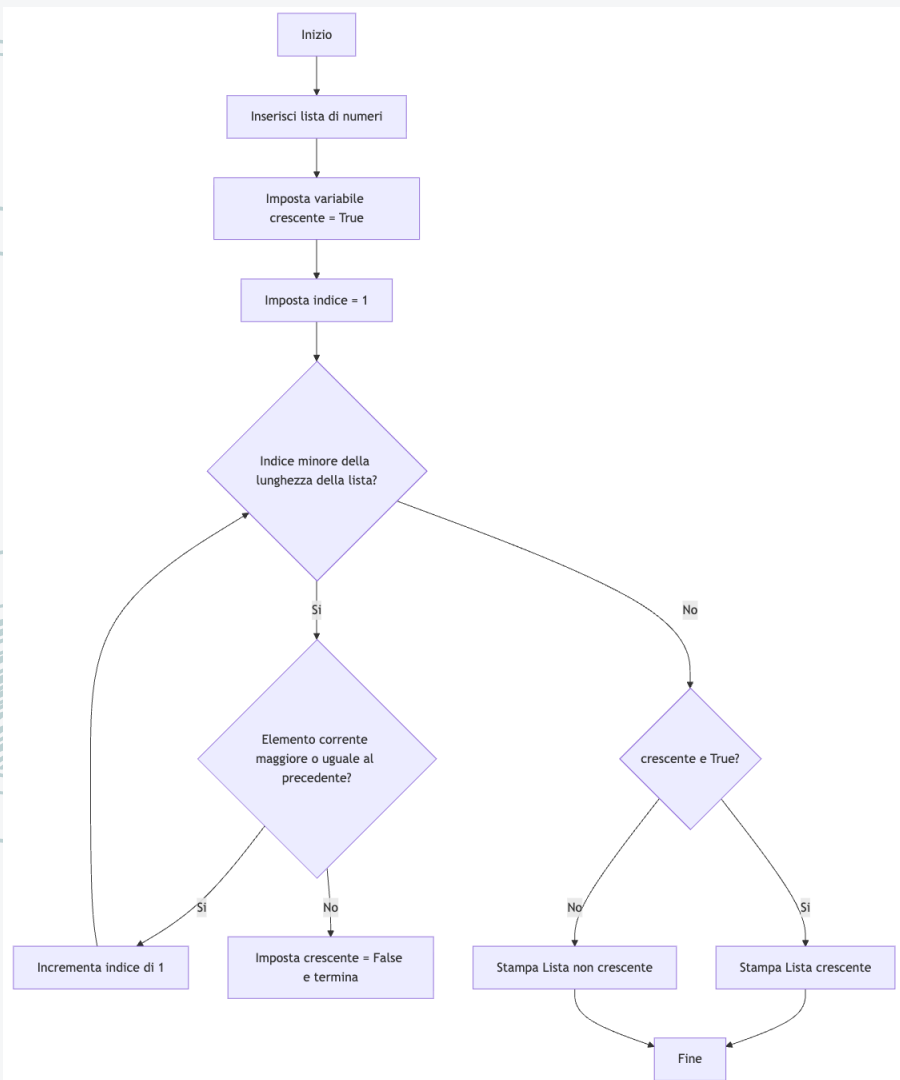
# Inizializza somma
somma = 0

# Ciclo sugli indici
for i in range(len(numeri)):
    if i % 2 == 0: # controlla se l'indice è pari
        somma += numeri[i]

# Stampa il risultato
print("La somma degli elementi agli indici pari è:", somma)

# Fine
```

VERIFICA SE UNA LISTA È CRESCENTE: SCRIVI UN PROGRAMMA CHE PRENDA UNA LISTA DI NUMERI E VERIFICHI SE LA LISTA È IN ORDINE CRESCENTE, OSSIA SE CIASCUN ELEMENTO È MAGGIORE O UGUALE ALL' ELEMENTO PRECEDENTE.



```
# Inizio

# Inserisci una lista di numeri
numeri = [1, 2, 2, 4, 5]

# Imposta variabile
crescente = True

# Controllo della lista
for i in range(1, len(numeri)):
    if numeri[i] < numeri[i - 1]:
        crescente = False
        break # basta un caso negativo per fermarsi

# Stampa il risultato
if crescente:
    print("La lista è crescente")
else:
    print("La lista NON è crescente")

# Fine
```

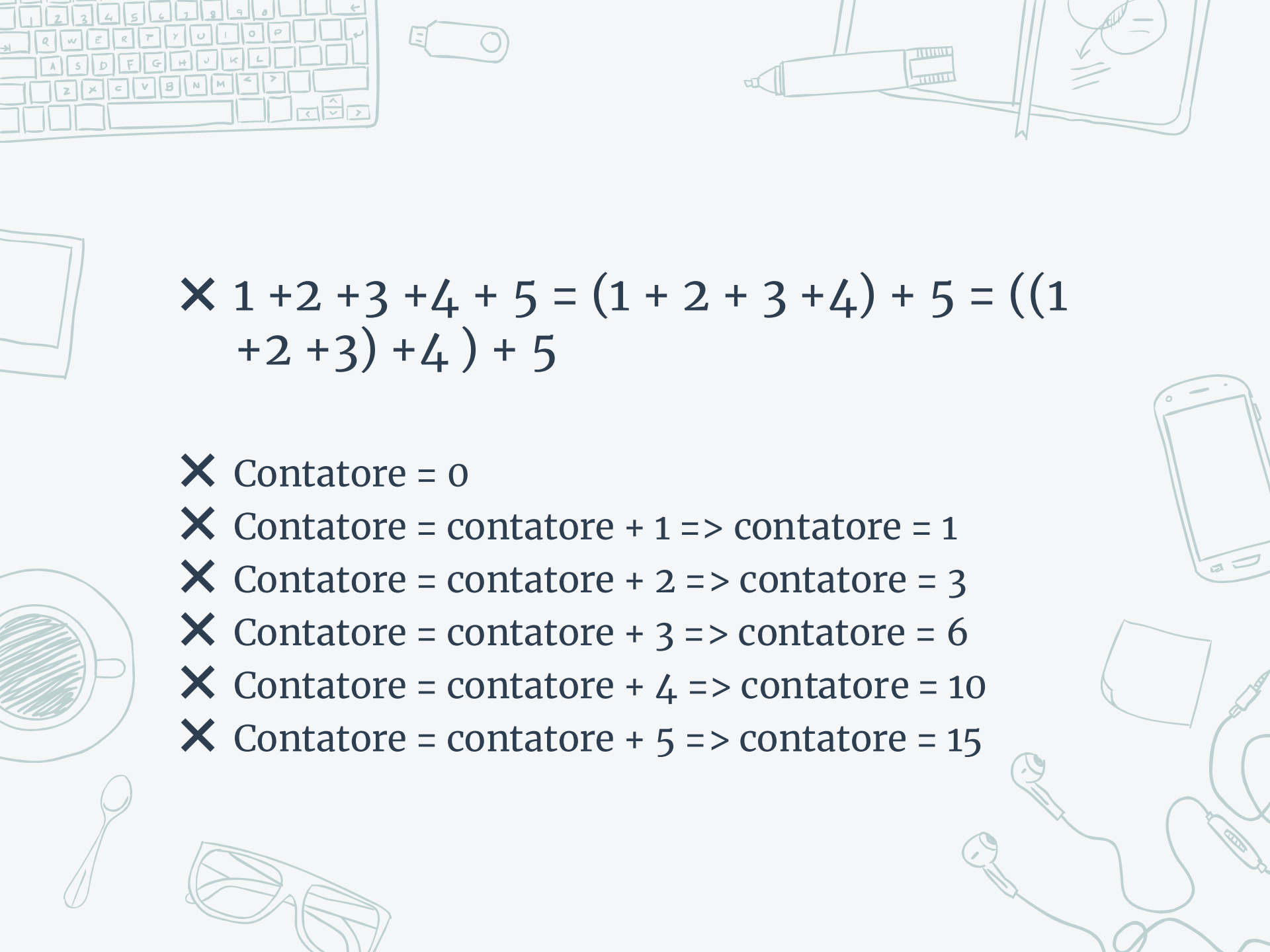
ESERCIZI DI RIPASSO



ESERCIZIO 6

✗ Dato un numero N determinare la somma dei primi numeri interi positivi minori o uguali ad N . Ad esempio se $N=5$ allora $(1+2+3+4+5=15)$ il programma dovrà stampare a video un messaggio del tipo:
"La somma dei numeri interi positivi minori o uguali di 5 è uguale a 15 "

1. Disegnare diagramma di flusso
2. Scrivere pseudocodice
3. Scrivere programma Python


$$\times 1 + 2 + 3 + 4 + 5 = (1 + 2 + 3 + 4) + 5 = ((1 + 2 + 3) + 4) + 5$$

$$\times \text{Contatore} = 0$$

$$\times \text{Contatore} = \text{contatore} + 1 \Rightarrow \text{contatore} = 1$$

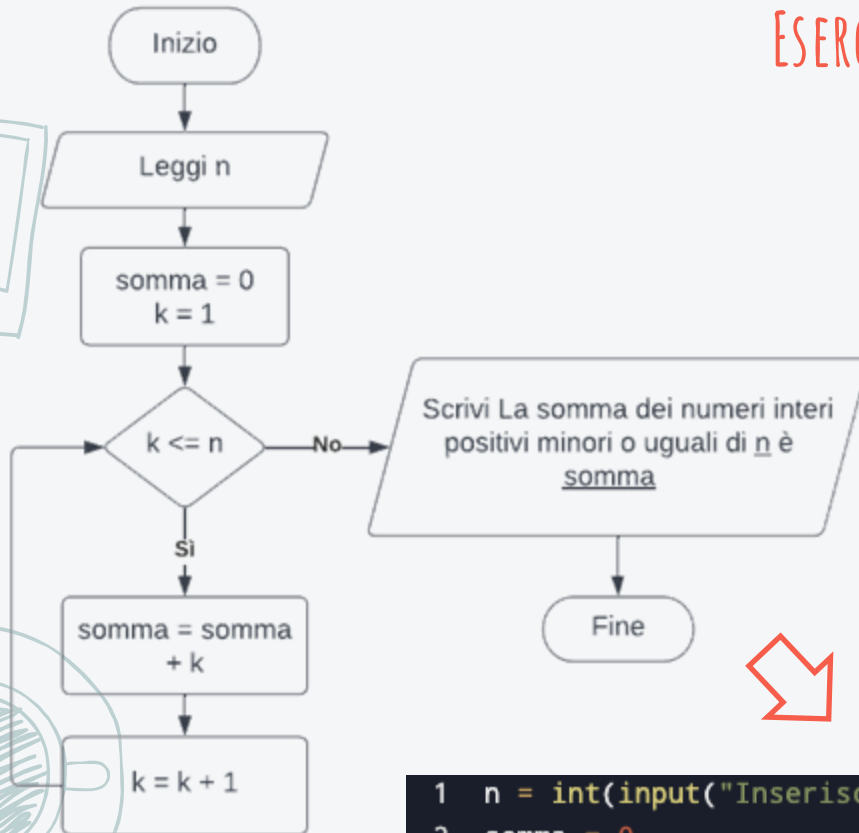
$$\times \text{Contatore} = \text{contatore} + 2 \Rightarrow \text{contatore} = 3$$

$$\times \text{Contatore} = \text{contatore} + 3 \Rightarrow \text{contatore} = 6$$

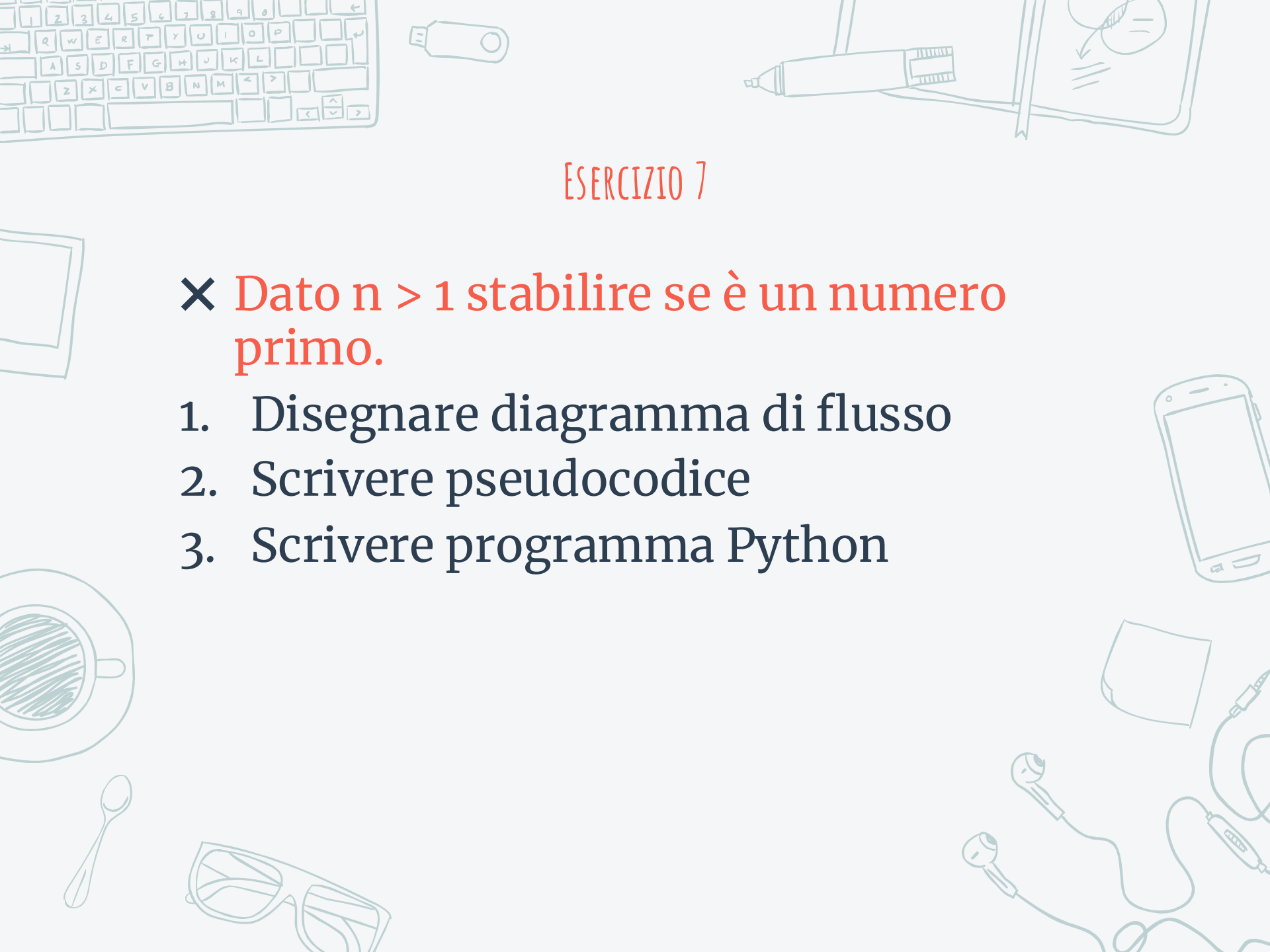
$$\times \text{Contatore} = \text{contatore} + 4 \Rightarrow \text{contatore} = 10$$

$$\times \text{Contatore} = \text{contatore} + 5 \Rightarrow \text{contatore} = 15$$

ESERCIZIO 6



```
1 n = int(input("Inserisci n: "))
2 somma = 0
3 k = 0
4 while k <= n:
5     somma = somma + k
6     k = k + 1
7 print(f"La somma dei numeri interi positivi minori o uguali di {n} è {somma}")
8 print("Fine")
```



ESERCIZIO 7

✗ Dato $n > 1$ stabilire se è un numero primo.

1. Disegnare diagramma di flusso
2. Scrivere pseudocodice
3. Scrivere programma Python

ESERCIZIO 2

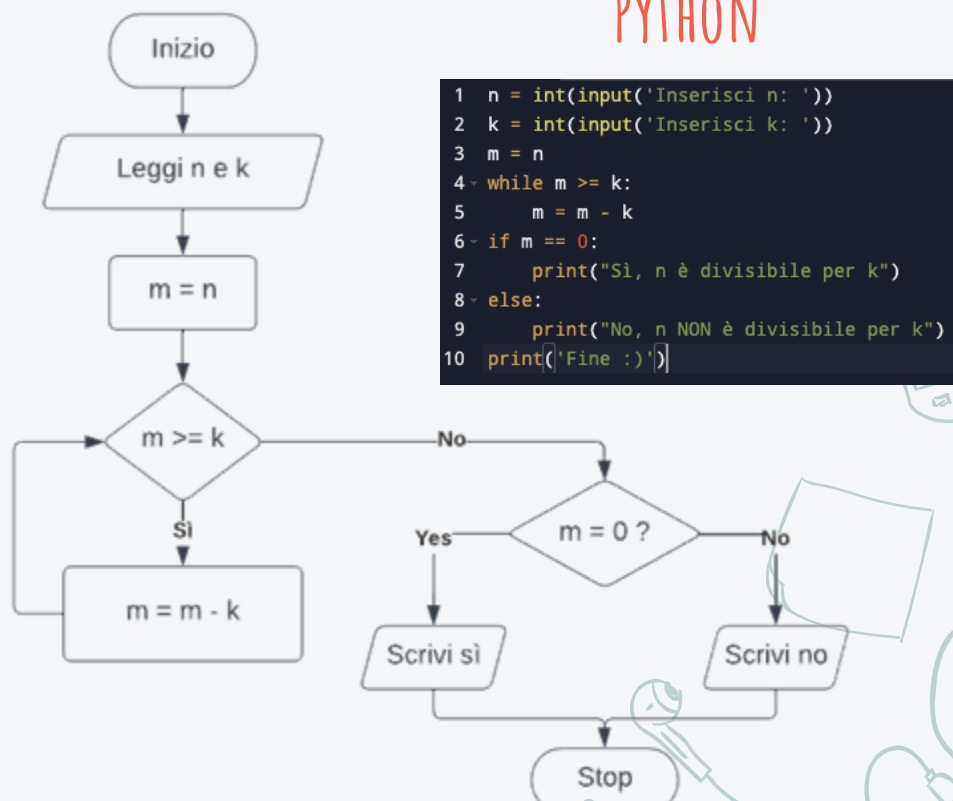


Dati $n, k > 0$ stabilire se n è divisibile per k .

La soluzione si basa sull'applicazione di una definizione assai elementare del concetto di divisibilità tra due numeri interi: possiamo dire che n è divisibile per k se k "entra" in n esattamente d volte; in altri termini n è divisibile per k se esiste un intero d tale che $n = kd$. La verifica avviene quindi sottraendo ripetutamente k da n fino ad ottenere il valore 0 nel caso in cui n sia divisibile per k o un numero maggiore di zero e minore di k , nel caso in cui n non sia divisibile per k .

```
1: leggi n e k
2: m = n
3: se m >= k allora prosegui,
   altrimenti vai al passo 6
4: m = m - k
5: vai al passo 3
6: se m = 0 allora scrivi "Sì, n è divisibile
   per k"
7: altrimenti scrivi "No, n non è divisibile
   per k"
8: stop
```

```
1: leggi n e k
2: m = n
3: fintanto che m >= k:
4:     m = m - k
6: se m = 0 allora
7:     scrivi "Sì, n è
   divisibile per k"
8: altrimenti
9:     scrivi "No, n non è
   divisibile per k"
10: stop
```



PYTHON

```
1 n = int(input('Inserisci n: '))
2 k = int(input('Inserisci k: '))
3 m = n
4 while m >= k:
5     m = m - k
6 if m == 0:
7     print("Sì, n è divisibile per k")
8 else:
9     print("No, n NON è divisibile per k")
10 print('Fine :)\n')
```

ESERCIZIO 7



Dato $n > 1$ stabilire se è un numero primo.

La soluzione si ottiene iterando per $k = 2, 3, \dots, n - 1$ l'algoritmo con cui si risolve il problema "stabilire se n è divisibile per k "

PYTHON

```
1 n = int(input("Inserisci n: "))
2 divisore = 2
3 while n % divisore != 0 and divisore < n:
4     divisore = divisore + 1
5 if divisore < n:
6     print("Il numero NON è primo")
7 else:
8     print("Il numero è primo")
9
10 print("Fine")
```

