Politecnico di Torino

Dipartimento
di Automatica e Informatica

Laboratorio 2

VARIABILI, OPERATORI ARITMETICI, STRINGHE

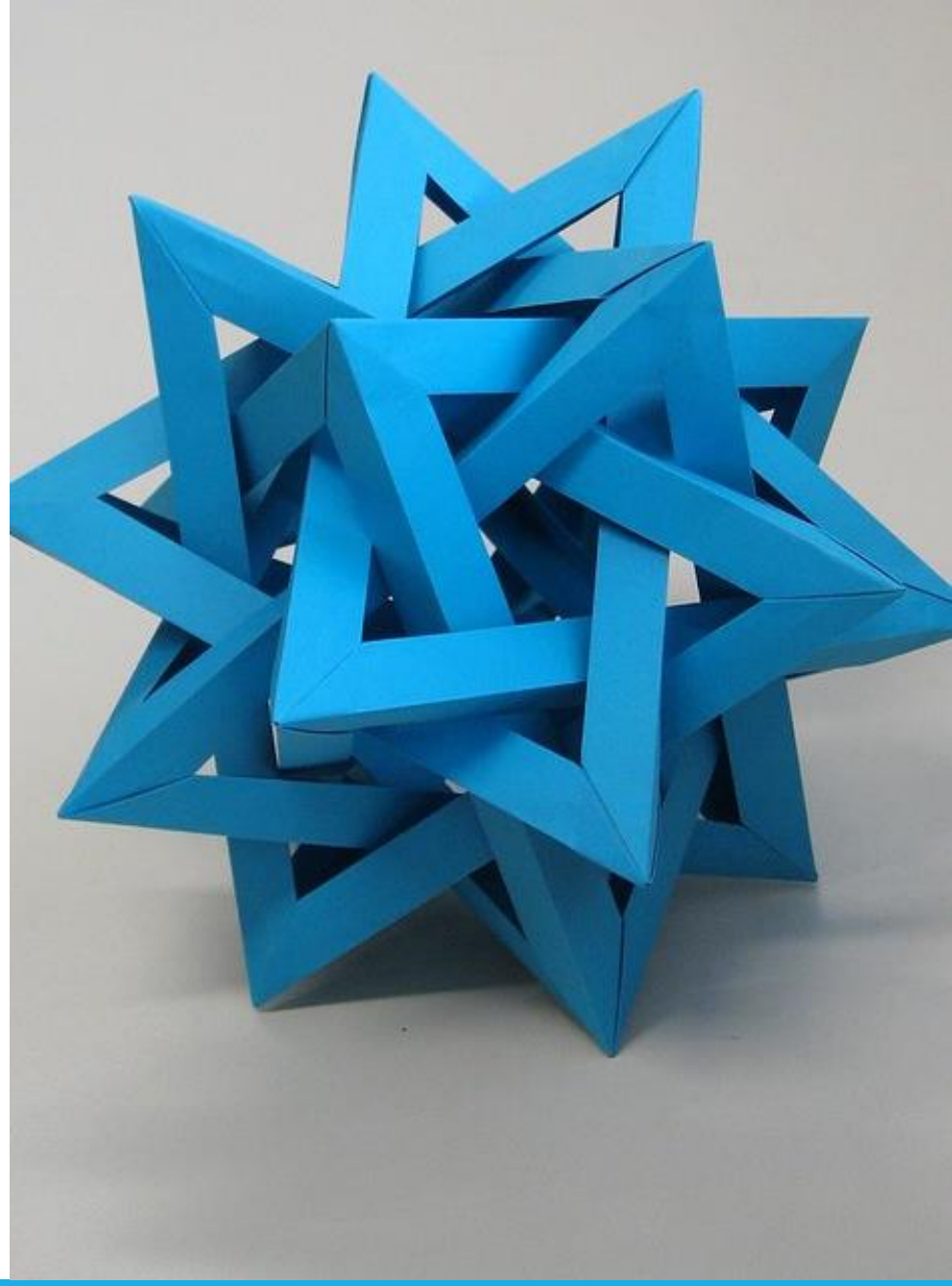
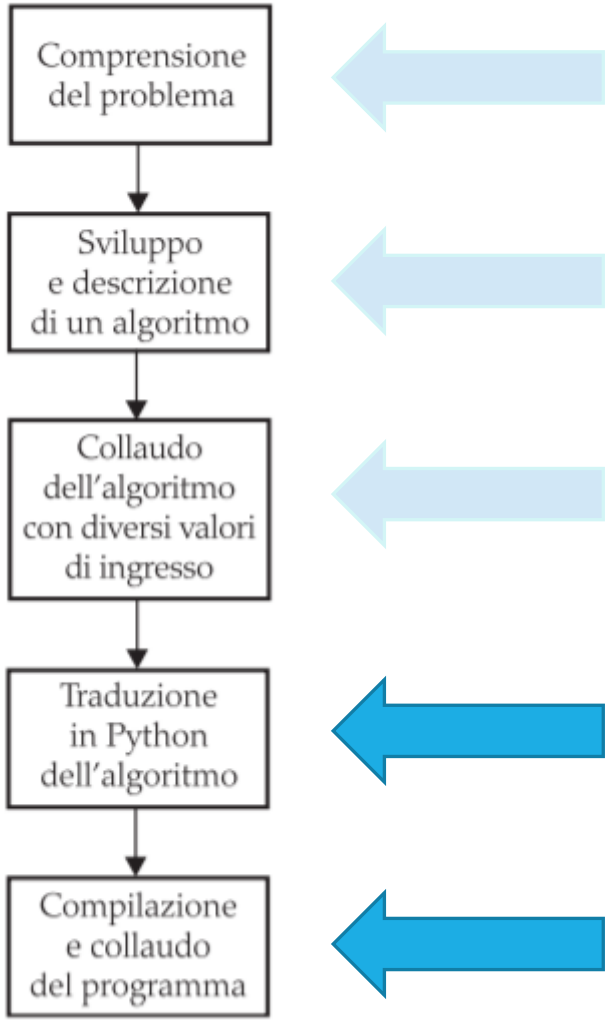


Figura 8

Procedimento per lo sviluppo di un programma



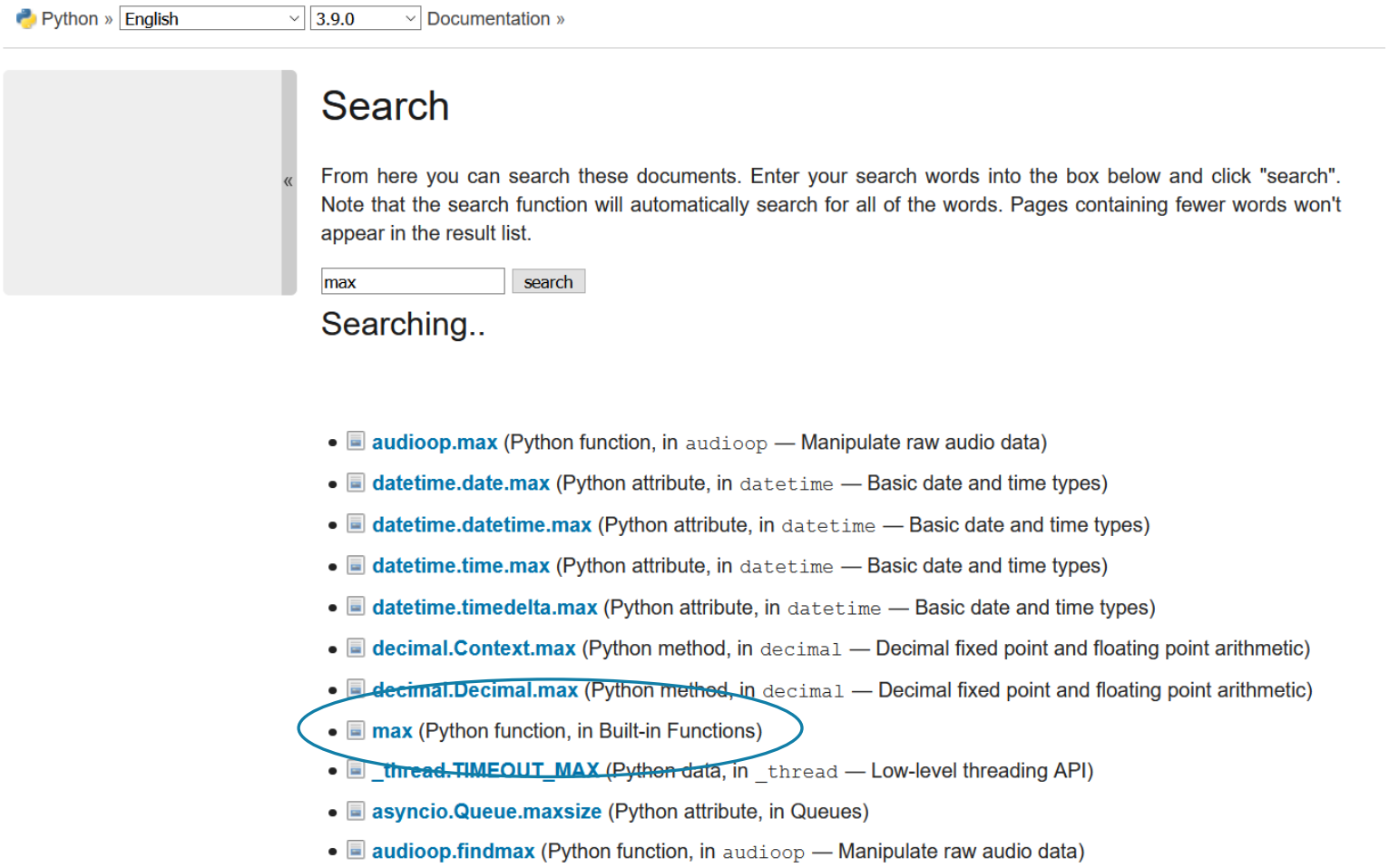
Esercizio 1

Esercizio 1. Scrivere un programma che memorizzi due numeri interi in due **costanti** definite nel codice, e poi ne visualizzi:

- La somma
- La differenza
- Il prodotto
- Il valore medio
- La distanza (cioè il valore assoluto della differenza)
- Il valore massimo (cioè il maggiore tra i due)
- Il valore minimo (cioè il minore tra i due)

Suggerimento: utilizzare le funzioni *max* e *min* definite in Python. Esse accettano una sequenza di valori separati da virgola in input e restituiscono rispettivamente il valore massimo e minimo della sequenza. (Es: `max(10, 5)` restituisce 10). [P2.4]

Esercizio 1 – *l'importanza della documentazione*



https://docs.python.org/3 /

Esercizio 1 – *l'importanza della documentazione*

max(*iterable*, *[, *key*, *default*])
max(*arg1*, *arg2*, **args*[, *key*])

Return the largest item in an iterable or the largest of two or more arguments.

If one positional argument is provided, it should be an iterable. The largest item in the iterable is returned. If two or more positional arguments are provided, the largest of the positional arguments is returned.

There are two optional keyword-only arguments. The *key* argument specifies a one-argument ordering function like that used for `list.sort()`. The *default* argument specifies an object to return if the provided iterable is empty. If the iterable is empty and *default* is not provided, a `ValueError` is raised.

If multiple items are maximal, the function returns the first one encountered. This is consistent with other sort-stability preserving tools such as `sorted(iterable, key=keyfunc, reverse=True)[0]` and `heapq.nlargest(1, iterable, key=keyfunc)`.

New in version 3.4: The *default* keyword-only argument.

Esercizio 1 – *il codice Python*

```python
# LAB2_es1
# Scrivere un programma che memorizzi due numeri interi in due costanti definite nel
# codice, e poi ne visualizzi:
# • La somma
# • La differenza
# • Il prodotto
# • Il valore medio
# • La distanza (cioè il valore assoluto della differenza)
# • Il valore massimo (cioè il maggiore tra i due)
# • Il valore minimo (cioè il minore tra i due)

A = 345
B = 23

# Calcoli
print("Somma:", A + B)
print("Differenza:", A - B)
print("Prodotto:", A * B)
print("Valor medio:", (A + B) / 2)
print("Distanza:", abs(A - B))
print("Massimo:", max(A, B))
print("Minimo:", min(A, B))
```

NB:

- Uso delle funzioni `min()` e `max()`
- Funzione `abs()`

Esercizio 2

Esercizio 2. Scrivere un programma che memorizzi in una costante un numero intero positivo *di cinque cifre*, e visualizzi le singole cifre di cui è composto.
Ad esempio, avendo il numero 16384, il programma deve visualizzare, su righe separate: 1 6 3 8 4. [P2.16]

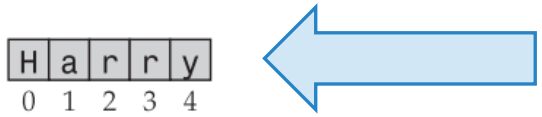
Esercizio 2 – *le stringhe*

2.4.4 Stringhe e caratteri

Le stringhe sono sequenze di caratteri **Unicode** (si veda la sezione Computer e società 2.1) ed è possibile accedere ai singoli caratteri contenuti in una stringa mediante la loro posizione al suo interno: tale posizione viene detta *indice* del carattere.

Il primo carattere ha indice 0, il secondo ha indice 1, e così via.

Le posizioni all'interno delle stringhe si contano a partire da zero.



Si può accedere a un singolo carattere di una stringa usando una speciale notazione a indice, racchiudendo la sua posizione fra una coppia di parentesi quadre. Se, ad esempio, la variabile `name` è definita in questo modo:

```
name = "Harry"
```

allora gli enunciati seguenti:

```
first = name[0]
last = name[4]
```

Esercizio 2 – *il codice Python*

```python
# LAB2_es2
# Scrivere un programma che memorizzi in una costante un numero intero positivo di
# cinque cifre, e visualizzi le singole cifre di cui è composto.
# Ad esempio, avendo il numero 16384, il programma deve visualizzare, su righe
# separate: 1 6 3 8 4


NUM1 = 16384 # costante
num = str(NUM1)

# oppure chiedere ad utente di inserire il numero
# num = input("Inserire un numero intero positivo di cinque cifre: ")

print(num[0])
print(num[1])
print(num[2])
print(num[3])
print(num[4])
```

Esercizio 3

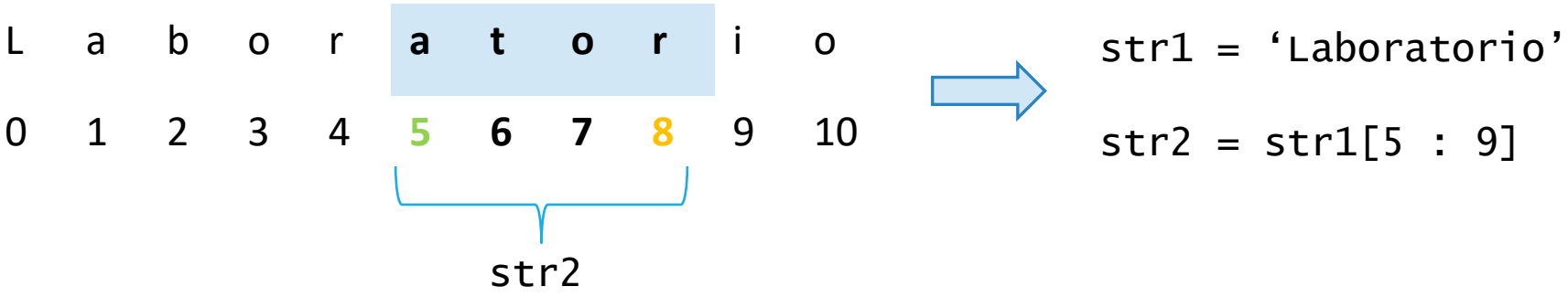
Esercizio 3. Scrivere un programma che memorizzi una stringa in una variabile e, a partire da quella variabile, visualizzi i primi tre caratteri della stringa, seguiti da tre punti, ancora seguiti dagli ultimi tre caratteri.
Ad esempio, se la stringa viene inizializzata al valore “Mississippi”, il programma deve visualizzare “Mis...ppi”. [P2.22]

Esercizio 3 – *slicing*

Lo **slicing** è l'operazione di estrazione di una sottostringa da una stringa.

La notazione è la seguente: `stringa[start : end : step]`

Esempio: estrarre la sottostringa '`ator`' dalla stringa '`Laboratorio`'



La sottostringa sarà estratta a partire dal carattere in posizione `start` della stringa, fino al carattere in posizione `end - 1` della stringa.

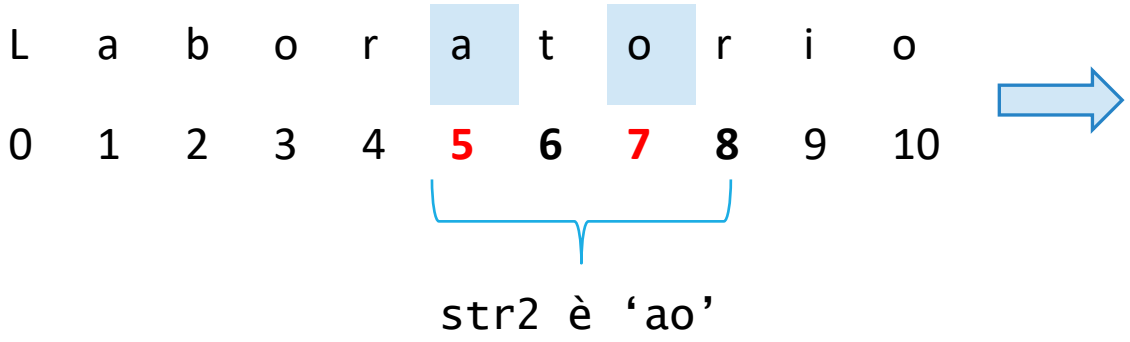
`step` può `essere` omesso, come in questo caso. Se omesso, viene considerato uguale a 1.

Esercizio 3 – *slicing*

Se `step` fosse stato 2 :

```
str1 = 'Laboratorio'

str2 = str1[5 : 9 : 2]
```



La sottostringa `str2` non è più 'ator', ma 'ao'.

Viene estratto, cioè, un carattere ogni due.

Esercizio 3 – *il codice Python*

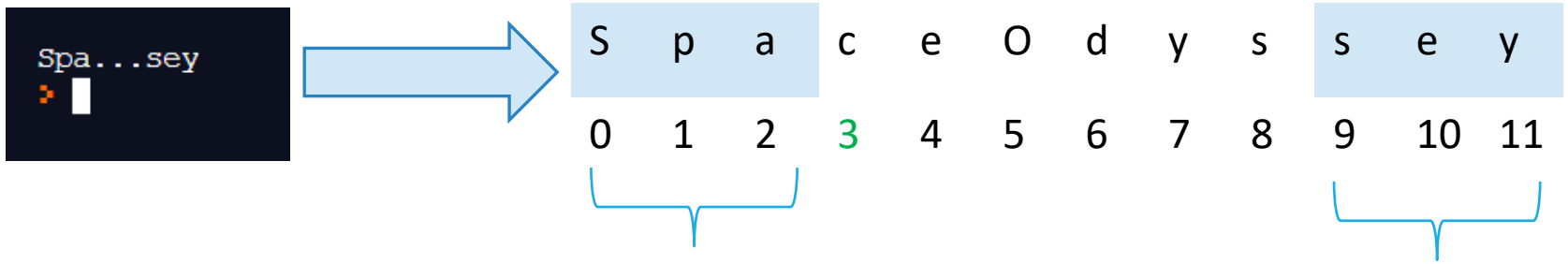
```python
## LAB2_es3
# Scrivere un programma che memorizzi una stringa in una variabile e, a partire da
# quella variabile, visualizzi i primi tre caratteri della stringa, seguiti da tre punti
# ancora seguiti dagli ultimi tre caratteri.

# Inizializzare la stringa
stringa = "SpaceOdyssey"

# Visualizza
print("%s...%s" % (stringa[0 : 3], stringa[len(stringa) - 3 : len(stringa)]))
```

Esercizio 3 – *esecuzione*

Eseguendolo si ottiene:




```
9   # Visualizza
10  print("%s...%s" % (stringa[0 : 3], stringa[len(stringa) - 3 : len(stringa)]))
```

12 – 3 = 9 12

NB:

- Si vede bene come con la notazione `stringa[0 : 3]` il carattere in *quarta* posizione (carattere 3), è escluso! Ricordare la notazione `string[start : end]` → la sottostringa viene estratta considerando gli indici start e end -1 della stringa.
- Uso della funzione `len()`

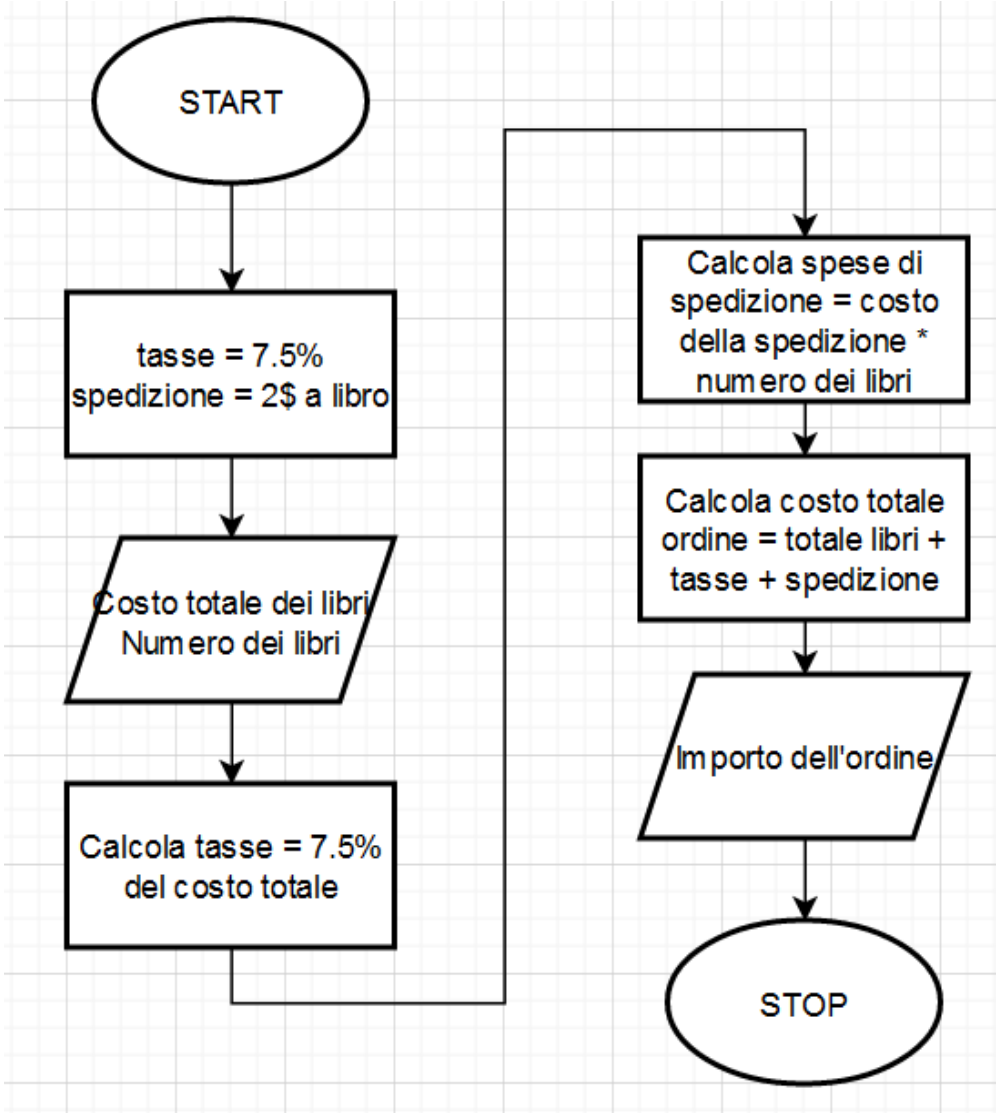
Esercizio 4

Esercizio 4. Lo pseudocodice seguente descrive come, in una libreria, viene calcolato l'importo di un ordine a partire dal costo totale dei libri ordinati e dal loro numero.

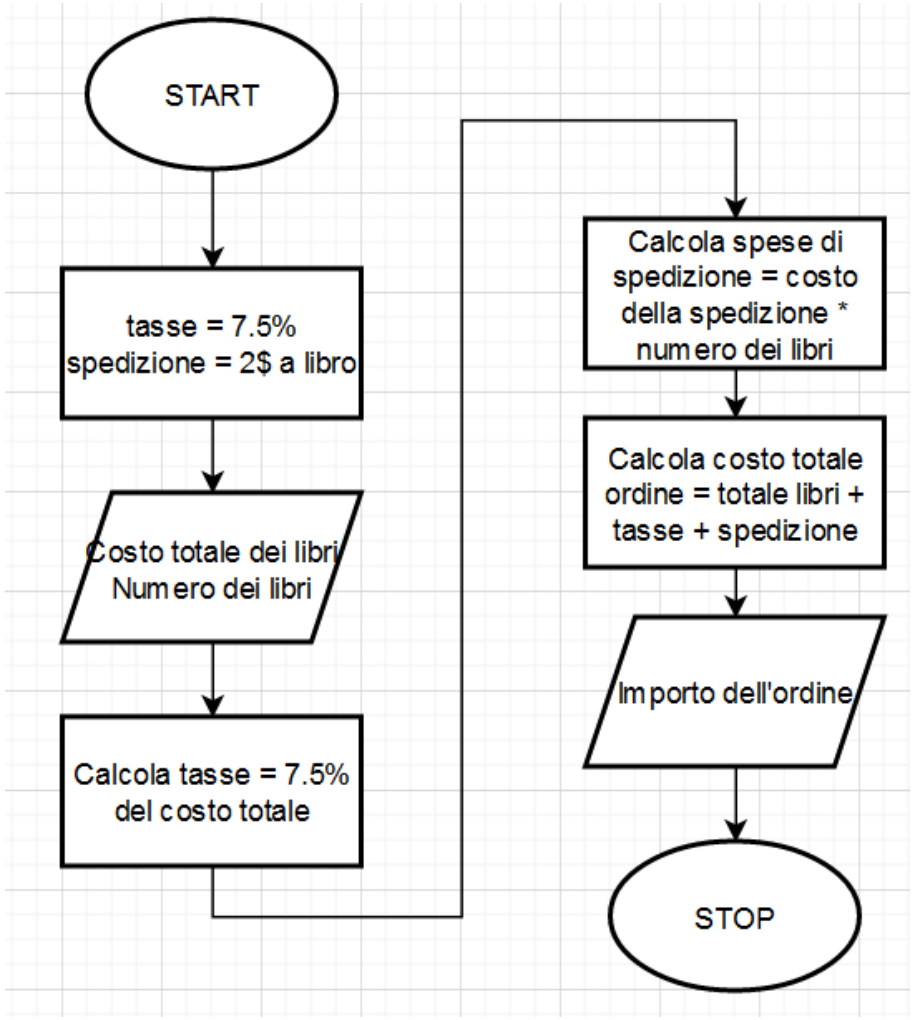
- *Leggere il costo totale dei libri e il numero di libri.*
- *Calcolare le tasse (il 7.5 per cento del costo totale dei libri).*
- *Calcolare i costi di spedizione ($2 per ogni libro).*
- *Il prezzo totale dell'ordine è la somma del costo totale dei libri, delle tasse e dei costi di spedizione.*
- *Visualizzare l'importo dell'ordine.*

Scrivere un programma in Python che implementi questo pseudocodice. Il costo totale dei libri e il numero di libri devono essere memorizzati in due variabili costanti. [P2.32]

Esercizio 4 – *flowchart*



Esercizio 4 – *definire variabili e costanti*



COSTANTI:
- TASSE
- SHIPPING_PER_LIBRO

VARIABILI:
- costoTotLibri
- numLibri
- tasseTot
- spedizioneTot
- costoTot

Esercizio 4 – *il codice Python*

```python
## LAB2_es4
#  Lo pseudocodice seguente descrive come, in una libreria, viene calcolato l'importo di
# un ordine a partire dal costo totale dei libri ordinati e dal loro numero.
# • Leggere il costo totale dei libri e il numero di libri.
# • Calcolare le tasse (il 7.5 per cento del costo totale dei libri).
# • Calcolare i costi di spedizione ($2 per ogni libro).
# • Il prezzo totale dell'ordine è la somma del costo totale dei libri, delle tasse e
# dei costi di spedizione.
# • Visualizzare l'importo dell'ordine.
# Scrivere un programma in Python che implementi questo pseudocodice. Il costo
# totale dei libri e il numero di libri devono essere memorizzati in due variabili
# costanti.
#


# Definisce le costanti
TASSE = 0.075
SHIPPING_PER_LIBRO = 2.0

# Input dall'utente
costoTotLibri = float(input("Inserire il costo totale dei libri: "))
numLibri = int(input("Inserire il numero di libri: "))

# Calcola il costo totale
tasseTot = costoTotlibri * TASSE
spedizioneTot = numLibri * SHIPPING_PER_LIBRO
costo_tot = costoTot + tasseTot + spedizioneTot

# Visualizza l'importo totale
print("Il costo totale dell'ordine è $%.2f." % costo_tot)
```

Esercizio 4 – *esecuzione*

Eseguendolo con dei valori d'esempio si ottiene:

```
Inserire il costo totale dei libri: 85
Inserire il numero di libri: 4
Il costo totale dell'ordine è $99.38.
>
```

NB:

- La notazione %.2f produce $99.38

`"%.2f"`	1 . 2 2	Visualizza due cifre dopo il punto decimale.

- L'uso di `float()` e `int()`