

Fondamenti di Informatica

Ing. Gestionale

a.a. 2022/2023

Esercitazione del 20/04/2023

Tra parentesi trovate (se esiste) il riferimento all'esercizio di riferimento (con possibili modifiche) dal libro di testo del corso:

Horstmann, C., & Necaie, R. D. (2019). *Concetti di Informatica e Fondamenti di Python* (2^a edizione). Maggioli Editore. ISBN: 9788891635433.

<http://www.apogeoeducation.com/concetti-di-informatica-e-fondamenti-di-python.html>

Esercizio 1 (modificato P5.21)

Scrivere una funzione che restituisca una lista contenente i primi n numeri dispari.

Alcuni esempi:

$n = 0 \rightarrow []$

$n = 1 \rightarrow [1]$

$n = 2 \rightarrow [1, 3]$

$n = 3 \rightarrow [1, 3, 5]$

Esercizi da svolgere a casa

Esercizio 2

Scrivere una funzione che data una matrice $m \times n$ (cioè con m righe e n colonne) restituisca una nuova matrice $(m + 1) \times (n + 1)$, in cui sia stata aggiunta una nuova colonna a destra che rappresenta la somma delle colonne ed una nuova riga sotto che rappresenta la somma delle righe.

Esempio:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 & 6 \\ 4 & 5 & 6 & 15 \\ 5 & 7 & 9 & 21 \end{pmatrix}$$

Nota: una matrice $m \times n$ può essere rappresentata come una lista di m elementi, ciascuno dei quali è a sua volta una lista di n elementi che rappresenta una riga.

Esercizio 3

Si scriva la funzione *ruota90gradiSensoOrario* che prenda in input un'immagine fatta di $n \times n$ caratteri disposti su n righe e n colonne e ritorni la stessa immagine ruotata di 90 gradi in senso orario (non preoccuparsi di girare le lettere 😊)

Esempio:

```
immagine = "XXXXX\n          output = ". . . . X\n              ..X..\n              ..X..\n              ..X..\n              ..X..\n              ..X.."          ". . . . X\n                              . . . . X\n                              XXXXX\n                              . . . . X\n                              . . . . X"
```

n = 5

Ricordarsi: l'accento vale un carattere '\n' e, per semplicità, l'ultima riga ha un accapo.

Esercizio 4

Scrivere una funzione che converta una stringa contenente una data dal formato "gg/mm/aaaa" al formato "aaaa/mm/gg".

Esercizio 5

Scrivere una funzione che data in input una lista di interi restituisca sia il numero di pari sia il numero di dispari.

Esercizio 6

Scrivere una funzione che data una lista restituisce True se e solo se la lista è ordinata in senso crescente (stretto)

Alcuni esempi:

[] → True

[1] → True

[1, 2] → True

[1, 2, 3] → True

[2, 1, 3] → False

[1, 1, 2] → False

Esercizio 7

Scrivere una funzione che data una lista di interi rimuova da essa (in place, senza farne una copia) tutti i numeri pari.

Esercizio 8

Scrivere una funzione che data una lista di interi restituisca True se e solo se ci sono solo numeri pari. In questo caso, a differenza di altre funzioni standard di Python viste in precedenza, si vuole che una lista vuota restituisca True.

Esercizio 9

Scrivere una funzione che data una lista di interi restituisca True se e solo se c'è almeno un numero pari.

Esercizi per casa della esercitazione precedente

Esercitazione 13/04: Esercizio 3

Le due varianti proposte differiscono nel fatto che si faccia o meno un controllo esplicito sulla lunghezza minima della stringa.

Esercitazione 13/04: Esercizio 4

Lo scopo dell'esercizio era quello di esercitarsi a scrivere funzioni Python. Infatti, le funzioni richieste esistono già in Python.

Per trovare la prima occorrenza di una stringa all'interno di un'altra si può usare il metodo *find* delle stringhe (che restituisce -1 se non trova alcuna occorrenza).

Per esempio,

```
"abc".find("b")
```

stampa

```
1
```

```
"abc".find("d")
```

stampa

```
-1
```

Il metodo *find* accetta un secondo argomento e opzionalmente un terzo argomento, che indicano la slice della stringa su cui effettuare la ricerca.

```
"abca".find("a", 1)
```

stampa

```
3
```

Per trovare l'ultima occorrenza, si può usare il metodo *rfind*

Non confondersi col metodo *index* che invece solleva un'eccezione se non trova un'occorrenza.

Per determina se tutti i caratteri di una stringa occorrono all'interno di un'altra stringa, vedremo (a breve) che si possono usare i *set*.