

Fondamenti di Informatica

Ing. Gestionale

a.a. 2022/2023

Esercitazione del 13/04/2023

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>

Appunti sparsi sul for:

Il costrutto *for* ci permette di iterare su una sequenza o altro oggetto *iterable* (es. le stringhe): per ciascun elemento della sequenza, la variabile associata al ciclo (a sinistra di *for*) viene inizializzata con l'elemento e quindi viene eseguito il corpo del ciclo:

```
stringa = "parola"  
for carattere in stringa:  
    print("carattere =", carattere)
```

Produce il seguente output:

```
carattere = p
carattere = a
carattere = r
carattere = o
carattere = l
carattere = a
```

Usando la "funzione" *reversed*, si può iterare a ritroso dall'ultimo elemento della sequenza al primo:

```
stringa = "parola"
for carattere in reversed(stringa):
    print("carattere =", carattere)
```

Produce il seguente output:

```
carattere = a
carattere = l
carattere = o
carattere = r
carattere = a
carattere = p
```

La "funzione" *range* può essere usata per iterare su una sequenza numerica:

```
for i in range(5):
    print("i =", i)
```

Produce il seguente output:

```
i = 0
i = 1
i = 2
```

```
i = 3
```

```
i = 4
```

Si noti che *range* con un solo argomento produce numeri interi a partire da zero, ad incrementi di uno e strettamente minori dell'argomento (che viene escluso).

La "funzione" *range* può essere invocata con due argomenti per indicare, con il primo argomento, il punto di partenza.

```
for i in range(2, 5):  
    print("i =", i)
```

Produce il seguente output:

```
i = 2
```

```
i = 3
```

```
i = 4
```

Con tre argomenti, si può indicare il "passo":

```
for i in range(1, 10, 2):  
    print("i =", i)
```

Produce il seguente output:

```
i = 1
```

```
i = 3
```

```
i = 5
```

```
i = 7
```

```
i = 9
```

Il "passo" può essere negativo: in questo caso, i valori nella sequenza prodotta devono essere strettamente maggiori del secondo argomento (che comunque viene escluso).

```
for i in range(9, 0, -2):  
    print("i =", i)
```

Produce il seguente output:

i = 10
i = 8
i = 6
i = 4
i = 2

Esercizio 1 (modificato P5.21)

Riscrivere il programma di conversione di un numero nella sua rappresentazione romana evitando la ripetizione di codice tra la gestione delle unità, decine e centinaia.

Esercizio 2

Ragionare su una funzione in grado di tradurre una parola o frase dalla lingua italiana al farfallese.

Parola → Pafarofolafa

Frase in input → Frafasefe ifin ifinpufut

È un gioco linguistico per bambini. In questo alfabeto le vocali vengono "codificate" per parlare in codice

La codifica consiste nel raddoppiare ogni vocale con l'aggiunta di una f interposta

Esercizi da svolgere a casa

Esercizio 3

Data una stringa in input, ragionare su una funzione in grado contare il numero di "x" in maniera tale che la lettera "x" appaia almeno 3 volte all'inizio o alla fine della stringa in input

X.....XX → 1 + 2 = 3

XXXX... → 4 + 0 = 4

XX...X → 2 + 1 = 3

X.....X → 1 + 1 = 2

....X.... → 0 + 0 = 0

Esercizio 4

Scrivere le seguenti funzioni:

1. Data una stringa s1 ed una stringa s2, restituisce l'indice della prima occorrenza di s2 in s1; altrimenti, restituisce -1 se non ve ne sono
2. Data una stringa s1 ed una stringa s2, restituisce l'indice dell'ultima occorrenza di s2 in s1; altrimenti, restituisce -1 se non ve ne sono
3. Data una stringa s1 ed una stringa s2, restituisce True se e solo se s1 è formata soltanto da caratteri presenti in s2 e s1 ha almeno un carattere

Esercizi per casa della esercitazione precedente

Esercitazione 06/22: Esercizio 3 (P4.1)

Scrivete programmi che, usando cicli, calcolino quanto segue:

- a. La somma di tutti i numeri pari compresi tra 2 e 100 (estremi inclusi);

- b. La somma di tutti i numeri compresi tra 1 e 100 (estremi inclusi) che siano quadrati perfetti;
- c. Tutte le potenze di 2, da 2^0 a 2^{20} ;
- d. La somma di tutti i numeri dispari compresi tra a e b (estremi inclusi), dove a e b sono valori acquisiti in ingresso;
- e. La somma di tutte le cifre dispari di un numero acquisito in ingresso (se, ad esempio, il numero è 32677, la somma da calcolare è $3 + 7 + 7 = 17$);

Esercizio 4

Scrivere un programma che legga in input una stringa s_1 ed una stringa s_2 e che quindi stampi tutte le posizioni in cui s_2 compare come sottostringa di s_1 .

ababaab ab -> 0 2 5

Commenti alle soluzioni:

L'idea generale è di spostarsi lungo la stringa s_1 e quindi verificare se da ciascuna posizione s_1 comincia con s_2 .

Commenti sulle varianti:

- a) Il ciclo while annidato serve a verificare se la stringa $s_1[i:]$ comincia con s_2 : si noti che si itera finché non si esaurisce s_2 o si è trovato un carattere non coincidente
- b) Usiamo l'operatore di slicing per estrarre da s_1 la sottostringa di $len(s_2)$ caratteri a partire dall'indice i e vedere se sia uguale a s_2

- c) Usiamo il metodo *startswith* delle stringhe per vedere se s_1 comincia con la stringa s_2 , partendo però dalla posizione i in s_1 .
- d) Il *while* esterno per iterare su s_1 è sostituito con un ciclo *for*
- e) Uso il metodo *find* delle stringhe per trovare una sottostringa data a partire da un determinato indice

Nota: la condizione $i \leq \text{len}(s_1) - \text{len}(s_2)$ è stata usata per escludere a monte le posizioni da cui non avevo abbastanza caratteri in s_1 per coprire tutta s_2 . In alternativa, sarebbe stato necessario progettare il ciclo interno in modo da verificare di non uscire dai limiti di ambo le stringhe: in realtà, metodi come lo *slicing*, *find* e *startswith* dovrebbero essere già abbastanza robusti per lo scopo.

Esercizio 5

Scrivere un programma che legga in input una stringa s_1 e stampi una stringa s_2 nella quale siano stati rimossi caratteri duplicati adiacenti.

Per esempio, la stringa *"aabbccad"* diventa *"abcad"*

Esercizio 6

Scrivere un programma che legga in input una stringa s_1 e stampi una stringa s_2 nella quale siano state inserite delle virgole ogni tre caratteri a partire dalla fine della stringa

Per esempio, la stringa *"12345679"* diventa *"12,345,679"*