

Fondamenti di Informatica
Corso di Laurea in Ingegneria Gestionale
A.A. 2017-18:
Esercitazione #4

Federico Scafoglieri
scafoglieri@diag.uniroma1.it

Domenico Lembo
lembo@diag.uniroma1.it

Obiettivo dell'esercitazione: Comprendere funzioni e liste.

Istruzioni

Scaricate il file "File_Utili_Esercitazione_04". all'interno troverete una cartella per esercizio contenente:

- Un file *TestEsN.py* che sarà il modulo da avviare per testare la propria soluzione;
- Un file *EsN.py* da completare implementando la funzione richiesta dall'esercizio;
- Un file *SolN.pyc* contenente la soluzione compilata, che sarà utile per la funzione di test.

Per portare a termine l'esercitazione aprite i file *EsN.py* e implementate la funzione seguendo la segnatura proposta nel testo dell'esercizio *N* di riferimento

Come testare il codice

Scrivete la funzione richiesta, delezionate la voce di menu "Run → Run Module" e nella shell inserite i parametri.

Per esempio data la seguente funzione:

```
1 def isOdd(number):  
2     if number % 2 == 0:  
3         return False  
4     else:  
5         return True
```

Run → Run Module

```
1 >>> isOdd(1)  
2 True  
3 >>> isOdd(2)  
4 False
```

Suggerimenti

Fate ATTENZIONE al tipo restituito dalla funzione. Se la funzione richiede di restituire un booleano *True/False*, restituire la stringa 'True'/'False' è un errore. Se si richiede di restituire la stringa 'error', restituire la stringa 'Error' è un errore.

Esercitazione Laboratorio

Esercizio 1

Difficoltà: ★

Scrivere la funzione Python *elementoiesimo*(*l*, *i*) che prende come parametri una lista *l* ed un intero *i* e restituisce l'elemento *i*-esimo di *l*. Se *i* eccede dai limiti della lista, la funzione deve restituire la **stringa** *error*

Esempio:

- Se $l = [a, f, r, d, g, e, q]$ e $i = 1$, *elementoiesimo*(*l*, *i*) restituisce *f*

Esercizio 2

Difficoltà: ★

Scrivere una funzione Python *uguali*(*x*, *y*) che prende come parametri due stringhe *x* e *y* e restituisce il **booleano** *True* se la stringa *x* in versione Upper Case è identica a *y*. Altrimenti restituisce il booleano *False*

Esempio:

- Se $x = \text{"ciao"}$ e $y = \text{"CIAO"}$, *uguali*(*x*, *y*) restituisce *True*
- se $x = \text{"ciao"}$ e $y = \text{"CIAOOOO"}$, *uguali*(*x*, *y*) restituisce *False*

Esercizio 3

Difficoltà: ★★

Scrivere una funzione Python *convertilista*(*l*, *i*, *j*) che prende come parametri una lista *l*, un intero positivo *i* e un intero positivo *j* e restituisce una **stringa** composta da soli '0' tranne che nella posizione *i*-esima e *j*-esima dove appariranno rispettivamente l'elemento *i*-esimo e *j*-esimo della lista.

Esempio:

- Se $l = [0, 1, 2, 3, 4, 5, 6]$ e $x = 1$ e $y = 3$, *convertilista*(*l*, *i*, *j*) restituisce '0103000'

Esercizio 4

Difficoltà: ★★★

Scrivere una funzione Python *contaoccorrenze*(*l*, *c*, *x*) che prende come parametri una lista *l*, un carattere *c* e un intero *x* e restituire il **booleano** *True* se il numero di occorrenze di *c* in *l* è maggiore o uguale a *x*, *False* altrimenti.

Esempio:

- se $l = ['a', 'a', 'v']$, $c = 'a'$ e $x = 1$, *contaoccorrenze*(*l*, *c*, *x*) restituisce *True*
- se $l = ['a', 'a', 'v']$, $c = 'z'$ e $x = 1$, *contaoccorrenze*(*l*, *c*, *x*) restituisce *False*

Esercizio 5

Difficoltà: ★★★

Scrivere una funzione Python *modificalista(l)* che prende come parametri una lista *l* e restituisce una **lista** ottenuta da *l* eliminando il primo elemento di *l* e sostituendo il secondo con uno 0. Verificate nella shell dei comandi il comportamento della funzione realizzata, e fate in modo che all'invocazione della funzione la lista passata come parametro NON venga modificata dopo la chiamata.

Esempio:

- se $l = [5, 2, 3, 4]$, *modificalista(l)* restituisce $[0, 3, 4]$, ma *l* è ancora uguale a $[5, 2, 3, 4]$

Esercizio 6

Difficoltà: ★★★

Scrivere una funzione Python *ritornalista(l1, l2)* che prende come parametri due liste di interi *l1* e *l2* della stessa lunghezza e restituisce una **lista** *lr* composta da due valori:

- $lr[0]$ è la sommatoria del prodotto elemento per elemento di *l1* e *l2*
- $lr[1]$ è il prodotto della sommatoria degli elementi della lista *l1* con la sommatoria degli elementi della lista *l2*

Esempio:

- se $l1 = [1, 3, 2]$ e $l2 = [1, 3, 2]$ il risultato sarà $[14, 36]$ poichè:
 $lr[0] = 1 * 1 + 3 * 3 + 2 * 2$
 $lr[1] = (1 + 2 + 3) * (1 + 2 + 3)$

Esercizio 7

Difficoltà: ★★★

Scrivere una funzione Python *verificadecrescenza(l1, l2)* che prende come parametri due liste di interi *l1* e *l2* della stessa lunghezza e restituisce il **booleano** *True* se ogni elemento della lista *l1* in posizione *i* è minore dell'elemento della lista *l2* in posizione *i*, ed è anche un suo multiplo, altrimenti restituisce *False* Esempio:

- se $l1 = [1, 3, 2]$ e $l2 = [2, 6, 8]$ restituisce *True*
- se $l1 = [1, 3, 15]$ e $l2 = [2, 6, 15]$ restituisce *False*
- se $l1 = [1, 3, 15]$ e $l2 = [2, 5, 30]$ restituisce *False*

Esercizio 8

Difficoltà: ★★★

Scrivere una funzione Python *contaoccorrenze(l, s)* che prende come parametri:

- una lista *l* contenente stringhe.
- una stringa *s*

e per ogni stringa st di l calcola il numero n di caratteri di s che compaiono in st , e restituisce una **nuova lista** in cui per ogni st in l , e nello stesso ordine di occorrenza di st in l , compaia prima st e poi n .

Esempio:

- se $l = ['abc', 'cde', 'bdd']$ e $s = 'bde'$ allora la funzione dovrà restituire $['abc', 1, 'cde', 2, 'ddf', 2]$

Esercizio 9

Difficoltà: ★★★

Scrivere una funzione Python *calcoladistanza*(l) che prende come parametro una lista l composta da coppie di punti $[x, y]$ e restituisce una **lista** $lret$ composta solo e soltanto dai punti che distano **almeno 3 unità** dal punto $[0, 0]$

- Se $l = [[15, 16], [1, 1], [10, 9]]$ restituisce $[[15, 16], [10, 9]]$

Nota: importare la libreria `math` per risolvere l'esercizio

HomeWork

Esercizio 10

Homework

Difficoltà: ★

Scrivere una funzione Python *estrai_lista(l, x, y)* che prende come parametri una lista l , un intero positivo x ed un intero positivo y , con $x < y$, e restituisce la **lista** nel range x, y .

Esempio:

- Se $l = [0, 1, 2, 3, 4, 5, 6]$ e $x = 0$ e $y = 3$ restituisce $[0, 1, 2]$

Esercizio 11

Homework

Difficoltà: ★

Scrivere tre funzioni Python:

- *eliminadispari(l)*, che prende come parametro una **lista di interi** l e restituisce una lista uguale ad l ma senza i numeri dispari
- *aggiungizero(l)*, che prende come parametro una **lista di interi** l e restituisce una lista uguale ad l ma a cui aggiunge in coda un numero di 0 pari al valore dell'ultimo elemento di l
- *modificalista(l)*, che prende come parametro una lista di interi l e restituisce una **lista di interi** uguale ad l ma senza numeri dispari ed a cui aggiunge in coda un numero di 0 pari al valore dell'ultimo elemento di l

Esercizio 12

Homework

Difficoltà: ★★★

Scrivere una funzione Python *moltiplicastringa(l, s)* che prende come parametri una lista l composta da numeri interi e una stringa s e restituisce una **stringa** composta dalla concatenazione delle stringhe risultanti da $s * l[i]$

Esempio:

- se $l = [1, 3, 2]$ e $s = 'cc'$ il risultato sarà $'cc' * 1 + 'cc' * 3 + 'cc' * 2 = ccccccccc$

Esercizio 13

Difficoltà: ★★★★★★

Scrivere tre funzioni:

- *stampatriangolo(base)* che prende come parametro un numero intero dispari ($base$) e stampa un triangolo isoscele di 'o' avente la base di lunghezza pari a $base$
- *stampaquadrato(lato)* che prende come parametro un intero ($lato$) e stampa un quadrato di 'o' con lunghezza del lato pari a $lato$

- *stamparettangolo(base, altezza)* che prende come parametri due interi (*base* e *altezza*) e stampa un poligono di 'o' di altezza pari ad *altezza* e lunghezza pari a *base*

Scrivere inoltre una funzione *main()* che chiede all'utente di inserire in input la tipologia di figura geometrica da stampare (1 per il triangolo, 2 per il quadrato, 3 per il rettangolo), richiede di inserire i parametri relativi, ed esegue la funzione richiesta.