

Esercitazione 3

10 novembre 2015

Termine per la consegna dei lavori: **martedì 17 novembre ore 23.59.**

Istruzioni

I lavori dovranno essere salvati in una cartella che deve contenere tutto e solo ciò che volete venga consegnato e valutato (generalmente sarà sufficiente un file di testo per ognuno degli esercizi).

Per consegnare gli elaborati dovete raggiungere la cartella contenente i file da inviare in modalità terminale (`cd path_della_cartella`) e quindi eseguire il comando:

```
consegna consegna3
```

verrà visualizzata la lista di tutto ciò che è stato inviato.

Consegne successive (entro il termine per la consegna) sovrascriveranno le precedenti, verrà valutata solo l'ultima consegna sottomessa.

È obbligatorio che all'interno di ogni file sia riportato il vostro nome, cognome e numero di matricola (potete riportarli all'interno di una riga commento all'inizio del file, es: `#Mario Rossi 1234567`).

ATTENZIONE!

In questa esercitazioni non sono ammesse importazioni di moduli esterni. Esercizi risolti utilizzando moduli importati riceveranno il punteggio minimo. Python contiene delle **built-in functions**, che potete utilizzare e la cui lista si può trovare a questo indirizzo: <https://docs.python.org/2/library/functions.html>. Si sottolinea, comunque, che questa esercitazione può essere risolta utilizzando **solo** funzioni viste in laboratorio e ogni esercizio deve funzionare con qualsiasi input che rispetti la consegna.

Consiglio generale: usate i descrittori di lista!

Esercizio 1

Sia $n > 0 \in \mathbb{N}$, un intero positivo, generare la lista di tutti e soli i numeri primi strettamente minori di n . Esempio:

```
>>> n = 20
>>> #vostro codice ...
>>> #risultato atteso
>>> [2, 3, 5, 7, 11, 13, 17, 19]
```

Esercizio 2

Creare una lista contenente le terne pitagoriche (a, b, c) con $a, b, c \in \mathbb{N}$ positivi e < 200 . Le terne dovranno apparire nella lista sotto forma di tupla a meno di permutazioni. L'ordine non è importante. Si ricorda che una terna pitagorica è una terna di numeri naturali a, b, c tali che $a^2 + b^2 = c^2$. La lista risultante dovrà essere della forma seguente:

```
>>> #vostro codice ...
>>> #risultato atteso
>>> [(3, 4, 5), (5, 12, 13), (6, 8, 10), (7, 24, 25), (8, 15, 17), ...]
```

Esercizio 3

Data una stringa di testo composta solo di caratteri alfabetici minuscoli e spazi, crearne l'“indice analitico” in questo modo: creare un dizionario contenente come chiavi tutte e sole le iniziali delle parole contenute nel testo e ad ognuna di queste dovrà essere associato come valore un altro dizionario contenente come chiavi le parole che iniziano con quella lettera e come valore il numero di occorrenze della parola nel testo.

Esempio:

```
>>> testo = "tre strette tazze dentro a tre tazze strette"
>>> #vostro codice ...
>>> #risultato atteso
>>> {'a': {'a': 1},
>>>  's': {'strette': 2},
>>>  't': {'tre': 2, 'tazze': 2},
>>>  'd': {'dentro': 1} }
```

Esercizio 4

Dato un intero $m > 0$ generare la lista di liste (**st**) contenente in posizione 0 la lista di tutti i numeri palindromi minori di m con unità pari a 1, in posizione 1 la lista di tutti i palindromi minori di m con unità pari a 2 e così via fino a 9. Creare poi altre due liste di liste (**sp**, **sd**), a partire da **st** contenenti rispettivamente, solo i palindromi pari e dispari. Si ricorda che: sia n un numero intero e sia $a_0, a_1, a_2, \dots, a_k$ la sua rappresentazione in cifre, allora n è palindromo se e solo se per ogni intero $0 \leq i \leq k$ si ha $a_i = a_{k-i}$, e.g, 12321 è palindromo (letto “al contrario” è sempre 12321), mentre non lo è 12323. Esempio:

```

>>> m = 200
>>> #vostro codice ...
>>> #risultato atteso
>>> st = [[1, 11, 101, 111, 121, 131, 141, 151, 161, 171, 181, 191],
>>>        [2, 22], [3, 33], [4, 44], [5, 55], [6, 66], [7, 77],
>>>        [8, 88], [9, 99]]
>>> sd = [[1, 11, 101, 111, 121, 131, 141, 151, 161, 171, 181, 191],
>>>        [3, 33], [5, 55], [7, 77], [9, 99]]
>>> sp = [[2, 22], [4, 44], [6, 66], [8, 88]]

```

Esercizio 5

Dato un testo in chiaro composto esclusivamente da caratteri alfabetici minuscoli (senza spazi) di **lunghezza pari** ed una chiave di cifratura k , scrivere uno script che implementa il cifrario di Hill. Il cifrario di Hill è un cifrario a sostituzione basato sull'algebra lineare. Nella funzione di cifratura, ogni lettera è per prima cosa codificata in numero: per semplicità codificheremo ogni lettera con la sua posizione all'interno dell'alfabeto: $a = 0, b = 1, c = 2, \dots, z = 25$. Consideriamo il testo suddiviso in blocchi di lunghezza 2 e codifichiamo le lettere come descritto sopra ottenendo dei vettori $\in \mathbb{Z}_{26}^2$: ad ognuno di questi si applica la cifratura con una chiave $\in \mathbb{Z}_{26}^{2 \times 2}$, semplicemente applicando la moltiplicazione matrice per vettore (modulo 26). Vediamo un semplice esempio: consideriamo il testo "help". I blocchi di lunghezza 2 che costituiscono questa stringa sono "he" e "lp". Applicando la codifica otteniamo i vettori:

$$\text{"he"} \rightarrow \begin{pmatrix} h \\ e \end{pmatrix} \rightarrow \begin{pmatrix} 7 \\ 4 \end{pmatrix}, \text{"lp"} \rightarrow \begin{pmatrix} l \\ p \end{pmatrix} \rightarrow \begin{pmatrix} 11 \\ 15 \end{pmatrix}.$$

Considerata la chiave $k \in \mathbb{Z}_{26}^{2 \times 2}$:

$$k = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix},$$

calcoliamo

$$\begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 7 \\ 8 \end{pmatrix},$$

$$\begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} 11 \\ 15 \end{pmatrix} = \begin{pmatrix} 0 \\ 19 \end{pmatrix}.$$

Da cui applicando la codifica dei caratteri all'indietro otteniamo:

$$\begin{pmatrix} 7 \\ 8 \end{pmatrix} \rightarrow \begin{pmatrix} h \\ i \end{pmatrix} \rightarrow \text{"hi"}, \begin{pmatrix} 0 \\ 19 \end{pmatrix} \rightarrow \begin{pmatrix} a \\ t \end{pmatrix} \rightarrow \text{"at"},$$

ottenendo la stringa cifrata "hiat". Si ricorda che le operazioni matrice per vettore sono state svolte **modulo 26**.

Quindi, data una chiave k sotto forma di lista di liste (definizione di matrice come vista in laboratorio) e una stringa t , restituire la sua cifratura secondo il cifrario di Hill, ad esempio:

```
>>> #matrice 2x2 scritta in formato lista di lista
>>> k = [[3,3],[2,5]]
>>> #stringa di lunghezza pari
>>> testo = "esempioditestoinchiaro"
>>> #vostro codice ...
>>> #risultato atteso
>>> cifrato = "ouwqrszrdhouveldbnyqpa"
```