

Esercitazione 4

18 novembre 2014

Termine per la consegna dei lavori: **martedì 25 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, d'ora in poi sarà **obbligatoriamente** un file **eseguitabile** con estensione `.py` per ognuno degli esercizi. Controllate che l'esecuzione del comando:

```
python <nome_file>.py
```

per ognuno degli esercizi produca l'output desiderato.

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 consegna4
```

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 (riportati all'interno di una riga commento all'inizio del file, es: `#Mario Rossi 1234567`).

ATTENZIONE!

Gli unici moduli importabili ammessi in questa esercitazione sono i moduli `math` e `random`. Esercizi risolti utilizzando altri 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>.

Esercizio 1

Un naturale $n \geq 2$ è detto *perfetto* se è uguale alla somma dei suoi divisori naturali, escluso se stesso.

Esempio: $6 = 1 + 2 + 3$ è il più piccolo numero perfetto.

Scrivere un programma che cerchi i primi 4 numeri perfetti visualizzandoli a video. Riuscite a trovare anche il quinto in tempi ragionevoli?

Esercizio 2

Scrivere un programma che visualizzi le prime 15 righe del triangolo di Tartaglia. Riuscite a spaziare opportunamente le cifre in modo da disporle ben incolonnate?

```

          1
         1 1
        1 2 1
       1 3 3 1
      1 4 6 4 1
     1 5 10 10 5 1
    1 6 15 20 15 6 1
```

http://it.wikipedia.org/wiki/Triangolo_di_Tartaglia

Esercizio 3

Scrivere un programma che generi casualmente 100 interi compresi tra 1 e 20 memorizzandoli in una lista, e successivamente visualizzi i numeri secondo l'ordine di generazione *senza ripetizioni*.

Esempio: se

```
v = [20, 1, 4, 5, 1, 1, 9, 2, 5, 11, ...]
```

allora si devono visualizzare i numeri

```
[20, 1, 4, 5, 9, 2, 11, ...]
```

Esercizio 4

Scrivere un programma che, presa in *input* una stringa, visualizzi a video il numero di parole inserite senza l'utilizzo della funzione `len()`.

N.B. Ogni parola si intende separata da un'altra da uno o più spazi (caratteri separati da sola punteggiatura senza spazi formano un'unica parola).

Esempio: se

```
s = "Questo esercizio e` semplice. Lo risolvo subito... si`, subito dopo mangiato!!"
```

allora il programma deve visualizzare il numero

```
11
```

Esercizio 5: Mastermind

Scrivere un programma che generi casualmente un codice composto da 4 cifre comprese tra 0 e 9, senza comunicare all'utente la sequenza ottenuta. A questo punto all'utente viene chiesto di fare dei tentativi per indovinare la sequenza generata (inserendo quindi in input 4 numeri compresi tra 0 e 9).

Dopo ogni tentativo, il programma fornisce degli aiuti comunicando:

1. il numero di cifre corrette al posto giusto, cioè le cifre del tentativo che sono effettivamente presenti nel codice al posto corretto.
2. il numero di cifre corrette al posto sbagliato, cioè le cifre del tentativo che sono effettivamente presenti nel codice, ma non nel posto corretto.

N.B. Non bisogna comunicare *quali* cifre sono giuste o sbagliate, ma solo *quante*.

L'utente vince la partita se riesce ad indovinare il codice entro un numero di tentativi predeterminati (solitamente 9, riuscite ad avvisare l'utente dei tentativi rimasti?).

<http://it.wikipedia.org/wiki/Mastermind>