



Funzioni

Corso Programmazione Python 2024
Modulo 4 - Esercizi

Luca Di Pietro Martinelli

Parte del materiale deriva dai corsi dei proff. Paolo Caressa e Raffaele Nicolussi (Sapienza Università di Roma) e Giorgio Fumera (Università degli Studi di Cagliari)

Esercizio 1

Si scriva del codice Python che definisca una funzione denominata `massimo`, che restituisca il più grande tra due numeri ricevuti come argomenti.

Richiedere poi all'utente tramite input da console due numeri (interi o frazionari) e stamparne il massimo richiamando la funzione appena definita.

Esercizio 2

Si scriva del codice Python che definisca una funzione denominata `massimo_comune_divisore`, che restituisca il massimo comune divisore di due numeri ricevuti come argomenti.

Richiedere poi all'utente tramite input da console due numeri interi e stamparne il massimo comune divisore richiamando la funzione appena definita.

Suggerimento: utilizzare l'algoritmo di Euclide.

Esercizio 3

Si crei un file Python in cui si definisca:

- una funzione denominata `is_numero_primo`, che prende in input un numero intero > 0 e restituisce `True` se tale numero è primo oppure `False` altrimenti
- una funzione denominata `stampa_numeri_primi`, che prende in input due numeri `a` e `b` e stampa tutti i numeri primi compresi tra `a` e `b`. Fare in modo che per verificare se un numero è primo si utilizzi la funzione `is_numero_primo`.

Scrivere poi in un secondo file Python del codice che richieda all'utente in input due numeri interi e che, richiamando le funzioni definite precedentemente, stampi tutti i numeri primi compresi tra questi due numeri.

Esercizio 4

Si crei un file Python in cui si definisca:

- una funzione denominata `controlla_alfanumerico`, che riceve in input una stringa e restituisce `True` se la stringa è alfanumerica (cioè contiene solo caratteri alfabetici o numerici) e `False` altrimenti
- una funzione denominata `stampa_controllo_alfanumerico`, che prende in input una lista di stringhe e stampa per ciascuna "Il valore è alfanumerico" se è alfanumerica e "Il valore non è alfanumerico" altrimenti.

Scrivere poi in un secondo file Python del codice che richieda all'utente in input cinque stringhe e che, richiamando le funzioni definite precedentemente, stampi se sono alfanumeriche oppure no.

Suggerimento: per controllare se un carattere è alfanumerico usare `in` e la stringa: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789".

Esercizio 5

Si crei un file Python in cui si definisca:

- una funzione denominata `genera_numeri_random`, che riceve in input un numero intero `dim` e genera una lista di lunghezza pari a `dim` contenente numeri interi casuali tra 1 e 100
- una funzione denominata `calcola_pari_dispari`, che prende in input un intero e restituisce la stringa "pari" se il numero è pari e la stringa "dispari" altrimenti.

Scrivere poi in un secondo file Python del codice che richieda all'utente in input un intero e che, richiamando le funzioni definite precedentemente, generi una lista con numeri random della lunghezza dell'intero fornito e per ciascun numero della lista stampi se è pari o dispari.

Esercizio 6

Si crei un file Python in cui si definisca:

- una funzione denominata `genera_numeri_random`, che riceve in input un numero intero `dim` e genera una lista di lunghezza pari a `dim` contenente numeri interi casuali tra 1 e 10
- una funzione denominata `fattoriale`, che prende in input un intero e restituisce il suo fattoriale (il fattoriale di `n`, indicato come `n!`, è pari a $n * n-1 * n-2 * \dots * 2 * 1$).

Scrivere poi in un secondo file Python del codice che richieda all'utente in input un intero e che, richiamando le funzioni definite precedentemente, generi una lista con numeri random della lunghezza dell'intero fornito e per ciascun numero della lista stampi il suo fattoriale.

Esercizio 7

Si crei un programma Python in cui si definisca una funzione

```
genera(n, a, b)
```

che, ricevendo in input tre numeri interi n , a e b , restituisce una lista con n numeri interi scelti a caso tra a e b (estremi inclusi).

La funzione deve controllare che i parametri siano interi positivi e che a sia minore di b .

Testare poi il comportamento della funzione appena creata.

Esercizio 8

Si crei un programma Python in cui si definisca una funzione

```
genera_distinti(n, a, b)
```

che, ricevendo in input tre numeri interi n , a e b , restituisce una lista con n numeri interi distinti tra loro scelti a caso tra a e b (estremi inclusi).

La funzione deve controllare che i parametri siano interi positivi e che a sia minore di b .

Testare poi il comportamento della funzione appena creata.

Esercizio 9

Si crei un programma Python in cui si definisca una funzione

```
is_distinct_int(list_int)
```

che, ricevendo in input una lista di numeri interi `list_int` restituisce `True` se tutti i numeri sono distinti tra loro.

La funzione deve controllare che i numeri siano tutti interi positivi.

Testare poi il comportamento della funzione appena creata creando la lista `list_int` ricavando i valori da input dell'utente.

Esercizio 10

Si crei un programma Python in cui si definisca una funzione

```
get_distinct_string(list_string)
```

che, ricevendo in input una lista di stringhe `list_string` restituisce la lista delle stringhe distinte.

La funzione deve controllare che i numeri siano tutti interi positivi.

Testare poi il comportamento della funzione appena creata creando la lista `list_string` ricavando i valori da input dell'utente.