

Esercizio 1

Si consideri il seguente programma Python:

```
def f2(m,i):
    v = 0
    for x in range(len(m)):
        v += m[x][i]
    return v

def f1(m):
    a = 0
    b = 0
    for i in range(len(m)):
        a += m[i][i]
        b += m[i][len(m) - 1 - i]
    if a != b:
        return False
    for i in range(len(m)):
        if a != sum(m[i]) or a != f2(m,i):
            return False
    return True

m = [ [7, 12, 1, 14],
      [2, 13, 8, 11],
      [16, 3, 10, 5],
      [9, 6, 15, 4] ]

print(f1(m))
```

Si descriva sinteticamente la funzione svolta dal programma e si mostri la traccia d'esecuzione, sapendo che l'output prodotto è *True*.

Esercizio 2

Si scriva una funzione *verifica_comuni* che riceve una lista *x* di stringhe e un intero *k*. La funzione restituisce una lista definita come segue:

- se le stringhe contenute in *x* hanno esattamente *k* lettere in comune, la lista restituita contiene tali lettere (in qualsiasi ordine);
- altrimenti, la lista restituita è vuota.

Esempio: Se $x = ['abcde', 'abdcf', 'bwawc', 'wcabdf']$, *verifica_comuni*(*x*,3) restituisce $['a', 'b', 'c']$, mentre *verifica_comuni*(*x*,2) restituisce una lista vuota.

Esercizio 3

Si vuole realizzare un'applicazione per l'analisi dei dati riguardanti vendite effettuate in un giorno tramite distributori automatici. I dati sulle vendite sono memorizzati in una matrice V avente 4 colonne, in cui la generica riga $[cod, tipo, inc_mat, inc_pom]$ indica che il distributore con codice cod , di tipo $tipo$, ha incassato inc_mat durante la mattinata e inc_pom durante il pomeriggio. Incassi pari a zero indicano il fatto che il distributore è stato guasto. L'ubicazione dei distributori è memorizzata in un dizionario C , in cui ad ogni città è associata la lista dei tipi di distributori ubicati in quella città.

Si scriva un modulo Python che metta a disposizione (almeno) le seguenti funzioni:

1. $distributori_simili(V, cod)$, che restituisce la lista dei codici dei distributori che:
 - sono dello stesso tipo di quello con codice cod ;
 - durante la mattinata hanno incassato la stessa cifra di quello con codice cod .
2. $affidabilita_tipi(V)$ che restituisce un dizionario in cui ad ogni tipo è associato l'indice di affidabilità di quel tipo; tale indice è definito come F/T , dove F è il numero di distributori di quel tipo che non hanno avuto guasti durante la giornata e T è il numero totale di distributori di quel tipo.
3. $statistiche(V, C)$, che restituisce un dizionario in cui ad ogni città è associata una lista $[tot_mat, tot_pom]$ dove tot_mat e tot_pom sono, rispettivamente, il totale incassato durante la mattinata e durante il pomeriggio dai distributori ubicati in quella città.
4. $elimina_sposta(V, C, città, tipo)$, che modifica V ed C in modo che riflettano le seguenti azioni (in ordine):
 - 1) eliminazione di tutti i distributori presenti nella città $città$;
 - 2) spostamento nella città $città$ di tutti i distributori di tipo $tipo$.

Esempio: Se $V =$

'D1'	'T3'	130.0	115.0
'D2'	'T2'	0.0	125.0
'D3'	'T3'	120.0	105.0
'D4'	'T3'	120.0	0.0
'D5'	'T1'	110.0	100.0
'D6'	'T1'	0.0	110.0
'D7'	'T2'	130.0	110.0
'D8'	'T3'	120.0	100.0

e $C =$

Chiave	Valore
'Rende'	['T1', 'T3']
'Cosenza'	['T2']

allora:

- $distributori_simili(V, 'D4')$ restituisce la lista $['D3', 'D8']$.
- $affidabilita_tipi(V)$ restituisce il dizionario:

Chiave	Valore
'T1'	0.5
'T2'	0.5
'T3'	0.75

- $statistiche(V, C)$ restituisce il dizionario:

Chiave	Valore
'Rende'	[600.0, 530.0]
'Cosenza'	[130.0, 235.0]

- $elimina_sposta(V, C, 'Rende', 'T2')$ modifica V e C come segue:

$V =$

'D2'	'T2'	0.0	125.0
'D7'	'T2'	130.0	110.0

$C =$

Chiave	Valore
'Rende'	['T2']
'Cosenza'	[]