

Esercizio 1

Si consideri il seguente programma Python:

```
def f(a):
    z = []
    for x in range(len(a)):
        s = True
        for j in range(len(a)//2):
            if a[j][x] != a[len(a)-j-1][x]:
                s = False
                break
        if s:
            z.append(x)
    return z

m = [ [3, 9, 3, 2, 7],
       [9, 0, 2, 1, 6],
       [4, 2, 9, 4, 3],
       [2, 0, 2, 8, 6],
       [6, 9, 3, 8, 7] ]

c = int(input('Inserisci un numero: '))
v = f(m)
print(c == len(v))
```

Si descriva sinteticamente la funzione svolta dal programma, si mostri la traccia d'esecuzione, si determini un valore da inserire in input affinché l'output prodotto sia *True* e si indichi il valore della variabile *v*.

Esercizio 2

Si scriva una funzione *verifica_presenza* che riceve due array *a* e *b*, e restituisce un array *v* della stessa lunghezza di *a*. In particolare, l'*i*-esimo elemento di *v* è calcolato come segue:

- se *a[i]* è uguale ad almeno un elemento di *b*, allora *v[i] = a[i]*;
- altrimenti, *v[i]* è uguale alla somma degli elementi di *a* con indice minore o uguale ad *i*.

Esempio: Se *a* = [5, 2, 3, 5, 9, 3] e *b* = [3, 5, 7], l'array restituito è *v* = [5, 7, 3, 5, 24, 3].

Esercizio 3

Si vuole realizzare un'applicazione per la gestione delle informazioni riguardanti biglietti per partite del campionato di calcio emessi da una biglietteria. Ogni cliente è identificato con il suo codice fiscale e può comprare un solo biglietto per partita. I dati relativi ai biglietti sono memorizzati in una matrice B avente 4 colonne, in cui la generica riga $[cf, s1, s2, sett]$ denota il fatto che il cliente con codice fiscale cf ha acquistato un biglietto nel settore dello stadio $sett$ (es. 'curva', 'tribuna', 'tribuna coperta', ecc.) per la partita che disputeranno le squadre $s1$ ed $s2$ nello stadio di $s1$ (la prima squadra è quella "di casa"). I prezzi dei biglietti sono memorizzati in un dizionario P in cui ad ogni squadra è associato un dizionario in cui, a sua volta, ad ogni settore è associato un prezzo.

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

1. $clienti_simili(B, cf1, cf2)$ che restituisce *True* se e solo se i clienti con codici fiscali $cf1$ e $cf2$ hanno assistito dallo stesso settore alla maggior parte delle partite (cioè a più della metà delle partite per le quali hanno acquistato un biglietto).
2. $spesa_media_clienti(B, P)$, che restituisce un dizionario in cui ad ogni cliente è associata la spesa media sostenuta, cioè la spesa totale sostenuta divisa per il numero di biglietti acquistati.
3. $statistiche(B, P)$, che restituisce un dizionario in cui ad ogni squadra che ha disputato almeno una partita in casa è associato l'incasso totale.
4. $clienti_vip(B, n)$, che restituisce la lista dei clienti *vip*. Un cliente è definito *vip* se soddisfa entrambe le seguenti condizioni:
 1. ha assistito ad almeno n partite;
 2. ha acquistato biglietti solo per il settore 'tribuna'.

Esempio: Se $B =$

'CF1'	'Milan'	'Inter'	'tribuna'
'CF2'	'Milan'	'Inter'	'curva'
'CF3'	'Milan'	'Inter'	'tribuna'
'CF4'	'Milan'	'Inter'	'curva'
'CF1'	'Juve'	'Napoli'	'curva'
'CF2'	'Juve'	'Napoli'	'curva'
'CF3'	'Juve'	'Napoli'	'tribuna'
'CF1'	'Inter'	'Lazio'	'curva'
'CF2'	'Inter'	'Lazio'	'curva'
'CF4'	'Inter'	'Lazio'	'tribuna'

e $P =$

Chiave	Valore
'Milan'	{ 'curva':30, 'tribuna':50 }
'Inter'	{ 'curva':30, 'tribuna':50 }
'Napoli'	{ 'curva':35, 'tribuna':45 }
'Juve'	{ 'curva':35, 'tribuna':60 }
'Lazio'	{ 'curva':25, 'tribuna':50 }

allora:

- $clienti_simili(B, CF1, CF2)$ restituisce *True* perché 'curva' è il settore da cui sia il cliente CF1 che il cliente CF2 hanno assistito alla maggior parte delle partite. In particolare: CF1 ha assistito a 3 partite, 2 dal settore 'curva' e una dal settore 'tribuna'; CF2 ha assistito a 3 partite, tutte dal settore 'curva'.

- $spesa_media_clienti(B, P)$ restituisce il dizionario:

Chiave	Valore
'CF1'	38.3
'CF2'	31.6
'CF3'	55
'CF4'	40

- $statistiche(B, P)$ restituisce il dizionario:

Chiave	Valore
'Milan'	160
'Inter'	110
'Juve'	130

- $clienti_vip(B, 2)$ restituisce la lista ['CF3'] perché il cliente con codice fiscale CF3 ha comprato solo biglietti di tribuna ed ha assistito ad almeno 2 partite, mentre tutti gli altri clienti, pur avendo assistito ad almeno 2 partite, hanno comprato biglietti sia di curva che di tribuna.