

Esercizio 1

Si consideri il seguente programma Python:

```
def f2(a,b,c):
    for i in range(len(a)):
        if a[i] / b[i] != c:
            return False
    return True

def f1(m,n):
    q = 0
    for i in range(1, len(m)):
        c = m[0][0] / m[i][0]
        if f2(m[0],m[i],c):
            q += 1
    return q == n

m = [ [1,2,1,4],
       [1,2,3,4],
       [4,1,7,3],
       [1,2,1,4],
       [2,4,2,8],
       [2,5,3,4] ]

n = int(input('Inserisci un numero: '))
print(f1(m,n))
```

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

Esercizio 2

Si scriva una funzione *verifica_lista* che riceve in ingresso una lista *L* di interi ed un intero *k*. La funzione restituisce *True* se e solo se la lista *L* non contiene nessuna sotto-lista di lunghezza *k* avente valori strettamente crescenti.

Esempio: Se $L = [4,5,5,6,1,1,2,2,5,3]$, allora *verifica_lista(L,3)* restituisce *True* perché nessuna delle sotto-liste di lunghezza 3 ($[4,5,5]$, $[5,5,6]$, $[5,6,1]$, $[6,1,1]$, $[1,1,2]$, $[1,2,2]$, $[2,2,5]$ e $[2,5,3]$) ha valori strettamente crescenti.

Esercizio 3

Si vuole realizzare un'applicazione per la gestione delle informazioni riguardanti acquisti di componenti per la produzione di articoli commerciali. L'applicazione memorizza i dati relativi alla composizione degli articoli in un dizionario *AC* in cui ad ogni articolo è associata la lista dei componenti necessari per la produzione dell'articolo stesso. Le offerte effettuate dai fornitori dei componenti sono memorizzate in una matrice *O* avente 3 colonne, in cui la generica riga $[comp, forn, pr]$ rappresenta il fatto che il componente *comp* è offerto dal fornitore *forn* al prezzo *pr*. Si assuma che non esistano due righe aventi uguali valori di *comp* e *forn* – in altri termini, ogni fornitore non può effettuare più offerte per uno stesso componente. La *spesa di fornitura* per un articolo è la somma dei prezzi che è necessario spendere per i componenti dell'articolo – ovviamente, se più di un fornitore fornisce uno stesso componente, si considera l'offerta con prezzo minore. Un componente è detto *speciale* quando è necessario alla produzione di un solo articolo.

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

1. *articolo_minima_spesa(AC,O)*, che restituisce l'articolo avente minima spesa di fornitura. Nel caso in cui più di un articolo soddisfi la condizione, la funzione ne restituisce uno qualsiasi.
2. *miglior_fornitore(AC,O)*, che restituisce il fornitore che soddisfa entrambe le seguenti condizioni:
 - a. il fornitore fornisce tutti i componenti necessari alla produzione di tutti gli articoli;
 - b. la somma dei prezzi richiesti dal fornitore è minima.

Nel caso in cui più di un fornitore soddisfi le condizioni, la funzione ne restituisce uno qualsiasi. Nel caso in cui nessun fornitore soddisfi le condizioni, la funzione restituisce una stringa vuota.

3. *componenti_speciali(AC)*, che restituisce la lista dei componenti speciali.
4. *componente_comune(O)*, che restituisce il componente disponibile presso il maggior numero di fornitori, cioè il componente per cui sono state effettuate il maggior numero di offerte. Nel caso in cui più di un componente soddisfi la condizione, la funzione ne restituisce uno qualsiasi.

Esempio: Se *AC* =

Chiave	Valore
'Art1'	['Comp1','Comp2']
'Art2'	['Comp2','Comp3']
'Art3'	['Comp4','Comp5']
'Art4'	['Comp4','Comp5']

e *O* =

'Comp1'	'Forn1'	100
'Comp1'	'Forn2'	70
'Comp2'	'Forn1'	120
'Comp2'	'Forn2'	100
'Comp2'	'Forn3'	70
'Comp3'	'Forn1'	90
'Comp3'	'Forn2'	90
'Comp3'	'Forn3'	80
'Comp4'	'Forn1'	70
'Comp4'	'Forn2'	70
'Comp5'	'Forn1'	90
'Comp5'	'Forn2'	100

allora:

- *articolo_minima_spesa(AC,O)* restituisce 'Art1', che è l'articolo avente minima spesa di fornitura (140).
- *miglior_fornitore(AC,O)* restituisce 'Forn2' perché:
 - 'Forn2' fornisce tutti i componenti necessari alla produzione di tutti gli articoli;
 - la somma dei prezzi richiesti da 'Forn2' è minima (430).
- *componenti_speciali(AC)* restituisce ['Comp1','Comp3'] perché 'Comp1' è necessario solo per 'Art1' e 'Comp3' è necessario solo per 'Art2'.
- *componente_comune(O)* restituisce 'Comp2' oppure 'Comp3', essendo questi i componenti disponibili presso il maggior numero di fornitori (3).