

Esercizio 1

Si consideri il seguente programma Python:

```
def f2(a,b):  
    x = sum(a[b])  
    for i in range(len(a)):  
        x -= a[i][b]  
    return x == 0  
  
def f1(x):  
    for i in range(len(x)):  
        if not f2(x,i):  
            return False  
    return True  
  
m = [[1,2,3,4],  
      [2,3,0,4],  
      [4,2,0,0],  
      [3,2,3,0]]  
  
print(f1(m))
```

Si descriva sinteticamente la funzione svolta dal programma e, in particolare, si mostrino la traccia d'esecuzione e l'output prodotto.

Esercizio 2

Si scriva una funzione f che riceve in ingresso due liste $L1$ ed $L2$ di stringhe e restituisce una lista che contiene gli elementi di $L1$ ed $L2$ in ordine crescente, senza elementi duplicati. La funzione non deve utilizzare la funzione *sort* della libreria standard.

Esempio: Se $L1=['albero', 'mare', 'zebra', 'casa']$ ed $L2=['zebra', 'albero', 'barca', 'verde', 'abaco']$, allora la funzione restituisce la lista $['abaco', 'albero', 'barca', 'casa', 'mare', 'verde', 'zebra']$.

Esercizio 3

Si vuole realizzare un'applicazione per la gestione delle notifiche di possibile contagio durante una pandemia. L'applicazione gestisce i possibili contagi in un ambiente di lavoro open-space; le postazioni in tale ambiente sono organizzate in una griglia, rappresentata da una matrice *posti* il cui generico elemento *posti[i][j]* contiene la matricola del dipendente che lavora abitualmente in posizione (i,j) all'interno della stanza, oppure -1 se la postazione è vuota. I dati relativi ai dipendenti sono memorizzati in un dizionario *dipendenti*, le cui chiavi sono le matricole dei dipendenti e in cui, a ciascun dipendente *x*, è associata una lista *dipendenti[x]* della forma *[nome, cognome, città, positivo]*, che rappresenta il fatto che il dipendente *nome cognome*, domiciliato nella città *città* è risultato positivo ad un tampone ed è quindi potenzialmente infettivo (se il booleano *positivo* ha valore *True*, altrimenti il dipendente non è risultato positivo).

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

1. *potenziale_infetto(posti,i,j,dipendenti,k)*, che restituisce *True* se e solo se nel posto di coordinate (i,j) c'è un dipendente che si trova a distanza minore o uguale a *k* da almeno un positivo, cioè se all'interno della sottomatrice con estremi $(i-k,j-k)$ e $(i+k,j+k)$ c'è almeno un collega positivo. Il parametro *k* è un numero naturale che chiamiamo *indice di contagiosità*; più è alto il valore di tale parametro, maggiore sarà la capacità del virus di contagiare persone distanti fra loro.
2. *dipendenti_potenziali_infetti(posti,dipendenti,k)*, che restituisce una lista contenente le matricole dei dipendenti che possono essersi infettati, se l'indice di contagiosità è *k*. La lista non deve contenere i dipendenti che sappiamo già essere positivi.
3. *infetti_città(posti,dipendenti,k)*, che restituisce un dizionario contenente, per ogni città (in cui risiede qualche dipendente), il numero di positivi o potenziali infetti, se l'indice di contagiosità è *k*.
4. *contagiosità_critica(posti,dipendenti)*, che restituisce il più piccolo indice di contagiosità tale che in ogni città (in cui risiede qualche dipendente) vi sia almeno un positivo o un potenziale infetto.

Esempio:

Se $\text{posti} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \\ L & M & N \end{bmatrix}$ e $\text{dipendenti} =$

Chiave	Valore
A	['marco','rossi','rende',True]
B	['carlo','verdi','rende',False]
C	['gino','neri','cosenza',False]
D	['franco','bianchi','cosenza',True]
E	['diego','viola','rossano',False]
F	['nino','russo','rossano',False]
G	['peppe','bruno','reggio',False]
H	['mario','chiari','reggio',False]
I	['ciro','ferrara','napoli',False]
L	['rino','sacchi','napoli',False]
M	['ugo','fantozzi','roma',False]
N	['andrea','piaggio','roma',True]

allora:

- *potenziale_infetto(posti,2,1,dipendenti,1)* restituisce *True* perché il dipendente *mario chiari* con matricola *H* in posizione $(2,1)$ è a distanza 1 dal dipendente *franco bianchi* con matricola *D* il quale risulta essere positivo.
- *dipendenti_potenziali_infetti(posti,dipendenti,1)* restituisce la lista $L=[B,E,G,H,M,I]$ perché i positivi *A*, *D* ed *N*, possono infettare i dipendenti $[B, E]$, $[B, E, H, G]$, e $[H, I, M]$, rispettivamente. Si noti che i dipendenti *A* e *D*, nonostante possano infettarsi a vicenda, non appartengono ad *L* perché risultano essere già infetti.

- *infetti_citta(posti,dipendenti,1)* restituisce il dizionario

Chiave	Valore
rende	2
cosenza	1
reggio	2
rossano	1
napoli	1
roma	2

perché: a *rende* vivono *A*, che è positivo, e *B* che è potenzialmente positivo perché a distanza 1 da *A*; a *cosenza* vivono *C*, che è potenzialmente positivo perché a distanza 1 da *D*, e *D* che è positivo; a *rossano* vivono *E*, che è potenzialmente positivo perché a distanza 1 da *D*, ed *F* che è negativo; a *reggio* vivono *G*, che è potenzialmente positivo perché a distanza 1 da *D*, ed *H* che è potenzialmente positivo perché a distanza 1 sia da *D* che da *N*; a *napoli* vivono *I*, che è potenzialmente positivo perché a distanza 1 da *N*, ed *L* che è negativo; a *roma* vivono *M*, che è potenzialmente positivo perché a distanza 1 da *N*, ed *N* che è positivo.

- *contagiosita_critica(posti,dipendenti)* restituisce 1 perché se $k=1$ allora in tutte le città esiste almeno un positivo o potenziale infetto.