

| | | | | | |
|------------------------|--|-------------------|--|---------------|--|
| Cognome e nome: | | Matricola: | | Corso: | <input type="checkbox"/> A <input type="checkbox"/> B |
|------------------------|--|-------------------|--|---------------|--|

Esercizio 1

Si consideri il seguente programma Python:

```
def f_2(m,c):
    for i in range(len(m)):
        for j in range(len(m)):
            if c and i == j and m[i][j] != 1:
                return False
            if not c and i != j and m[i][j] != 0:
                return False
    return True

def f_1(m):
    return f_2(m,True) and f_2(m,False)

m = [ [1,0,0,0],
       [0,2,0,0],
       [0,0,1,1],
       [0,0,0,1] ]

print(f_1(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 *elabora_lista* che riceve una lista *L1* di interi positivi e restituisce una lista *L2* della stessa lunghezza di *L1*, il cui generico elemento *L2[i]* contiene il massimo tra *spi* ed *ssi*, dove *spi* è la somma degli elementi di *L1* in posizione precedente a *i* ed *ssi* è la somma degli elementi di *L1* in posizione successiva a *i*. Ovviamente, se *i* = 0 si ha *spi* = 0, mentre se *i* = *len(L1)*-1 si ha *ssi* = 0.

Esempio: Se *L1* = [7, 4, 7, 3, 6, 8], allora la funzione restituisce la lista *L2* = [28, 24, 17, 18, 21, 27].

Esercizio 3

Si vuole realizzare un'applicazione per la gestione delle informazioni riguardanti prenotazioni di prestazioni professionali.

Le prenotazioni sono memorizzate in una matrice P avente 5 colonne, la cui generica riga $[giorno, ora, tipo, cognome, pagata]$ rappresenta il fatto che il cliente con cognome $cognome$ (per semplicità assumiamo che i clienti siano identificati univocamente dal proprio cognome) ha effettuato una prenotazione per una prestazione di tipo $tipo$ per il giorno $giorno$ e l'ora ora (questi ultimi rappresentati per semplicità come interi); $pagata$ è un booleano che indica se la prestazione è stata pagata. Le righe della matrice P non hanno alcun ordinamento specifico.

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

1. $unico_tipo(P, cognome, tipo)$, che restituisce *True* se e solo se tutte le prenotazioni effettuate dal cliente con cognome $cognome$ sono di tipo $tipo$.
2. $statistiche(P)$, che restituisce una matrice avente 2 colonne, la cui generica riga $[cognome, num_pren]$ rappresenta il fatto che il cliente con cognome $cognome$ ha effettuato num_pren prenotazioni.
3. $cliente_max(P)$, che restituisce il cognome del cliente che ha effettuato il maggior numero di prenotazioni. Nel caso in cui esistano più clienti con tale caratteristica, la funzione restituisce uno qualunque di essi. Nella scrittura di questa funzione, si suggerisce di richiamare la funzione di cui al punto 2.
4. $aggiungi_prenotazione(P, giorno, ora, tipo, cognome)$, che aggiunge una riga $[giorno, ora, tipo, cognome, False]$ alla matrice P se e solo se sono soddisfatte entrambe le seguenti condizioni:
 - non è stata già effettuata una prenotazione per lo stesso giorno e la stessa ora;
 - tutte le prenotazioni già effettuate dal cliente sono state pagate.

Esempio: Se $P =$

| | | | | |
|---|----|-----------------|-----------|-------|
| 1 | 13 | 'Progettazione' | 'Bianchi' | False |
| 1 | 9 | 'Consulenza' | 'Rossi' | True |
| 2 | 14 | 'Perizia' | 'Neri' | True |
| 3 | 9 | 'Progettazione' | 'Neri' | True |
| 3 | 10 | 'Consulenza' | 'Bianchi' | False |
| 3 | 12 | 'Consulenza' | 'Neri' | True |
| 1 | 10 | 'Perizia' | 'Neri' | True |
| 4 | 14 | 'Consulenza' | 'Rossi' | False |
| 2 | 11 | 'Progettazione' | 'Neri' | True |
| 4 | 15 | 'Progettazione' | 'Bianchi' | False |

allora:

- $unico_tipo(P, 'Rossi', 'Consulenza')$ restituisce *True*, perché tutte le prenotazioni effettuate dal cliente con cognome 'Rossi' sono di tipo 'Consulenza'.
- $statistiche(P)$ restituisce la matrice $\begin{bmatrix} 'Bianchi' & 3 \\ 'Rossi' & 2 \\ 'Neri' & 5 \end{bmatrix}$.
- $cliente_max(P)$ restituisce 'Neri', perché il cliente con cognome 'Neri' ha effettuato il maggior numero di prenotazioni (5).
- $aggiungi_prenotazione(P, 4, 16, 'Perizia', 'Neri')$ aggiunge una riga $[4, 16, 'Perizia', 'Neri', False]$ alla matrice P , perché
 - non è stata già effettuata una prenotazione per il giorno 4 alle ore 16;
 - tutte le prenotazioni già effettuate dal cliente 'Neri' sono state pagate.