Esercitazione di Laboratorio Informatica@DSS (2024/2025)

Massimo Lauria

Laboratorio 8 - 25/11/2024

Lo scopo del laboratorio è di esercitarsi e misurare la propria preparazione: gli esercizi non sono troppo difficili, se si sono seguite le lezioni. Non vi viene comunque messo alcun voto.

Modalità di lavoro: gli studenti devono lavorare in autonomia o in piccoli gruppi, senza disturbare i colleghi. Il lavoro di gruppo è fruttuoso solo se tutti partecipano e se ognuno scrive una propria soluzione per tutti gli esercizi.

Il docente cercherà per quanto possibile di non occupare il tempo del laboratorio per introdurre materiale nuovo, anche se a volte questo sarà necessario. Il docente è a disposizione per aiutare gli studenti, che possono iniziare a lavorare anche prima che il docente arrivi in aula, se lo desiderano

Raccomandazioni: leggete bene il testo degli esercizi prima di chiedere chiarimenti. In ogni caso sarò in aula con voi.

Uso dei file di test: per aiutarvi a completare questa esercitazione avete a disposizione dei programmi di test per testare la vostra soluzione. Questi sono simili a quelli che avrete in sede di esame, pertanto vi consiglio di imparate ad usarli. Per portare a termine l'esercizio è necessario

- scrivere un file di soluzione **col nome specificato**;
- avere dentro la funzione **col nome specificato**;
- porre il file di test corrispondente **nella stessa cartella**;
- eseguire in quella cartella il comando `python3 <fileditest>`

dove `<fileditest>` va ovviamente sostituito con il nome del file di test appropriato per l'esercizio su cui state lavorando. Per ogni esercizio ci sta un file di test indipendente, così da poter lavorare sugli esercizi uno alla volta con più agio.

Il risultato di ogni test è una schermata (o più schermate) nella quale si mostra:

- se è stato trovato il file con il nome corretto,
- se il file contiene la funzione con il nome corretto,
- se chiamate alla funzione con diversi valori dei parametri terminano restituendo risultati corretti.

Per ogni funzione scritta vengono eseguite chiamate con diversi valori dei parametri. L'esito dei test viene riportato con il carattere

- `E` se la chiamata non può essere eseguita,
- `F` se la funzione non restituisce il risultato corretto,
- `.` se la funzione restituisce il risultato corretto.

1 Stampare una matrice

Scrivere una funzione `stampamatrice(m)` che:

- riceva come parametro una matrice `m`;
- stampi la matrice `m`, sotto forma di tabella, andando a capo alla fine di ciascuna riga. Ad esempio per una matrice

  ```
  [ [2, 4, 3, 6, 8], [3, 1, 4, 0, 0], [1, 5, 1, 7, 4] ]
  ```

  deve stampare

  ```
  2 4 3 6 8
  3 1 4 0 0
  1 5 1 7 4
  ```

- la funzione non deve restituire nulla.

Per questo esercizio non viene fornito un file di test. Provare a chiamare la funzione con argomenti diversi e verificare l'aspetto della stampa.

2 Somma degli elementi di una matrice numerica

Scrivere una funzione `somma_mat(m)` che riceva come parametro una matrice `m` e restituisca la somma di tutti gli elementi della matrice. Ad esempio se riceve in come argomento la rappresentazione in python della matrice

$$\begin{bmatrix} 3 & 1 & 8 & 2 \\ 8 & -2 & 2 & 0 \\ 4 & -5 & 1 & 1 \end{bmatrix}$$

la funzione deve ottenere il valore 23.

Ricordiamo che in Python noi rappresentiamo una matrice $r \times c$ come una lista di lunghezza r, contenente r liste tutte quante di lunghezza c. Potete sempre assumere che l'input di questo esercizio sia una matrice ben formata, con $r \geq 1$ e $c \geq 1$, e che tutte le liste che rappresentano le righe contengano c valori numerici.

Il programma Python deve essere salvato nel file: `somma_mat.py`

File di test: `test_somma_mat.py`

3 Minimo degli elementi di una matrice numerica

Scrivere una funzione `min_mat(m)` che:

- riceva come parametro una matrice `m`;
- restituisca il valore dell'elemento minimo nella matrice.

Ricordiamo che in Python noi rappresentiamo una matrice $r \times c$ come una lista di lunghezza r, contenente r liste tutte quante di lunghezza c. Potete sempre assumere che l'input di questo esercizio sia una matrice ben formata, con $r \geq 1$ e $c \geq 1$, e che tutte le liste che rappresentano le righe contengano c valori numerici.

Il programma Python deve essere salvato nel file: `min_mat.py`

File di test: `test_min_mat.py`

4 Dizionario da stringa

Scrivete una funzione `dict_da_stringa_parole(stringa)` che:

- si aspetti come argomento una stringa costituita da un numero pari di parole, separate da spazi;
- crei e restituisca un `dict`, ottenuto prendendo le parole di posto pari (iniziando da 0) come chiavi, e le parole di posto dispari come valori.

Una stringa `s` può essere spezzata in porzioni in corrispondenza degli spazi attraverso il metodo `s.split()`, che restituisce una lista in cui ciascun elemento è una delle porzioni della stringa `s` delimitate da spazi.

Ad esempio, l'istruzione

`d = dict_da_stringa_parole("dog cane cat gatto mouse topo")`
deve assegnare a `d` il dict `{'dog': 'cane', 'cat': 'gatto', 'mouse': 'topo'}`.

Se la stringa non contiene alcuna parola la funzione deve restituire un dict vuoto.

```
d = dict_da_stringa_parole("dog cane cat gatto mouse topo")
print(d)
d = dict_da_stringa_parole("123 numero_grande 8 numero_piccolo")
print(d)
```

```
{'dog': 'cane', 'cat': 'gatto', 'mouse': 'topo'}
{'123': 'numero_grande', '8': 'numero_piccolo'}
```

Il programma Python deve essere salvato nel file: `dict_da_stringa_parole.py`

File di test: `test_dict_da_stringa_parole.py`

5 Somma degli elementi della diagonale di una matrice numerica

Scrivere una funzione `sommadiagonale_mat(m)` che:

- riceva come parametro una matrice `m`;
- restituisce la somma degli elementi sulla diagonale principale. Notare che per una matrice con `n` righe ed `n` colonne si devono sommare solo `n` elementi.

Ad esempio se riceve in come argomento la rappresentazione in python della matrice 3×3

$$\begin{bmatrix} 3 & 1 & 6 \\ 8 & -2 & 2 \\ 4 & -5 & 4 \end{bmatrix}$$

la funzione deve ottenere il valore 5.

Ricordiamo che in Python noi rappresentiamo una matrice $r \times c$ come una lista di lunghezza r, contenente r liste tutte quante di lunghezza c. Potete sempre assumere che l'input di questo esercizio sia una matrice ben formata, con $r \geq 1$ e $c \geq 1$, e che tutte le liste che rappresentano le righe contengano c valori numerici.

Il programma Python deve essere salvato nel file: `sommadiagonale_mat.py`

File di test: `test_sommadiagonale_mat.py`

6 Somme per riga di una matrice numerica

Scrivere una funzione `somme_per_riga_mat(m)` che:

- riceva come parametro una matrice `m` di numeri;
- restituisca una lista in cui l'elemento di posto `i` è la somma degli elementi della riga `i`-esima. La lunghezza di tale lista è uguale al numero di righe della matrice.

Ad esempio dalla rappresentazione della matrice

$$\begin{bmatrix} 3 & 1 \\ 8 & -2 \\ 4 & -5 \end{bmatrix}$$

la funzione deve ottenere la lista `[4, 6, -1]`.

Ricordiamo che in Python noi rappresentiamo una matrice $r \times c$ come una lista di lunghezza r, contenente r liste tutte quante di lunghezza c. Potete sempre assumere che l'input di questo esercizio sia una matrice ben formata, con $r \geq 1$ e $c \geq 1$, e che tutte le liste che rappresentano le righe contengano c valori numerici.

Il programma Python deve essere salvato nel file: `somme_per_riga_mat.py`

File di test: `test_somme_per_riga_mat.py`