

Fondamenti di Informatica

Prova d'esame del 30/06/2016

Regole d'esame:

1. È vietato parlare con altri studenti
2. È vietato consultare appunti, dispense, libri, in qualunque formato
3. È vietato tenere accesi i telefoni cellulari

Esercizio

Sia dato un testo T e due parole w_1 e w_2 presenti nel testo T . Diciamo che esiste una *co-occorrenza* di w_1 e w_2 in T se w_1 e w_2 appaiono consecutivamente in T (in qualunque ordine). La *matrice di co-occorrenza* di T è una matrice quadrata C che ha una riga e una colonna per ogni parola presente nel testo T e il cui elemento $C[i][j]$ indica il numero di co-occorrenze della parola i e della parola j . Ad esempio, La matrice di co-occorrenza del testo "Sopra la panca la capra campa sotto la panca la capra crepa" è la seguente:

	sopra	la	panca	capra	campa	sotto	crepa
sopra	0	1	0	0	0	0	0
la	1	0	4	2	0	1	0
panca	0	4	0	0	0	0	0
capra	0	2	0	0	1	0	1
campa	0	0	0	1	0	1	0
sotto	0	1	0	0	1	0	0
crepa	0	0	0	1	0	0	0

La classe **MatriceCoOccorrenze** rappresenta la matrice di co-occorrenza di un testo. Tale matrice è rappresentata mediante l'array di **String parole** che contiene tutte le parole presenti nel testo e mediante l'array bidimensionale di interi **coOccorrenze** il cui elemento (i,j) contiene il numero di co-occorrenze di **parole[i]** e **parole[j]**. Con riferimento all'esempio precedente si avrebbero i seguenti array:

sopra	0	1	0	0	0	0	0
la	1	0	4	2	0	1	0
panca	0	4	0	0	0	0	0
capra	0	2	0	0	1	0	1
campa	0	0	0	1	0	1	0
sotto	0	1	0	0	1	0	0
crepa	0	0	0	1	0	0	0

parole

coOccorrenze

Figura 1

```
public class MatriceCoOccorrenze {
```

```
    private String[] parole;        // array di parole
```

```
    private int[][] coOccorrenze;   // matrice dei valori di coOccorrenza
```

```
    /* Crea un oggetto MatriceCoOccorrenze il cui array di parole è p e la cui matrice dei valori di co-occorrenza è c */
```

```
    public MatriceCoOccorrenze (String[] p, int[][] c){...}
```

```

    /* Restituisce la parola i-esima */
    public String parola (int i){...}

    /* Riceve come parametro una parola p e restituisce la parola che ha il massimo numero di co-
    occorrenze con p. Se p non è una parola presente nell'array parole il metodo restituisce null */
    public String parolaMaxCoOccorrenze (String p){...}

    /* Restituisce un array di int il cui i-esimo elemento è il massimo numero di co-occorrenze della
    parola i-esima */
    public int[] maxCoOccorrenze (){...}

    /* Restituisce una descrizione testuale dell'oggetto MatriceCoOccorrenze su cui è invocato il metodo
    */
    public String toString (){...}

}

```

A titolo di esempio si immagini che **m** sia un oggetto di tipo **MatriceCoOccorrenze** rappresentato dagli array di Figura 1. L'invocazione **m.parolaMaxCoOccorrenze("la")** restituisce "panca". L'invocazione **m.maxCoOccorrenze()** restituisce l'array {1,4,4,2,1,1,1} in quanto il massimo numero di co-occorrenze per la parola "sopra" è 1, il massimo numero di co-occorrenze per "la" è 4, e così via.

Si scriva la classe **MatriceCoOccorrenze** ed una classe **ProvaMatriceCoOccorrenze** che contiene il solo metodo **main** e che esegue le seguenti azioni:

- Crea un oggetto **MatriceCoOccorrenze m** il cui contenuto è scelto dall'utente;
- Fa inserire ripetutamente una stringa **s** all'utente e dice all'utente la parola che ha più co-occorrenze con **s**, o un opportuno messaggio se **s** non fa parte delle parole di **m**; l'utente decide se continuare ad inserire nuove stringhe o andare avanti.
- Mostra all'utente, per ogni parola di **m**, il massimo numero di co-occorrenze.

Note:

1. Tutti i file creati devono essere salvati nella cartella C:\fi0630 del proprio PC.
2. Indicare il proprio cognome e nome, la propria matricola e il numero del PC su cui si sta lavorando negli appositi spazi in cima a questo foglio ed anche come commento in testa alla definizione delle classi che si scrivono su file.