

Fondamenti di Informatica

Prova d'esame del 25/6/2013

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

Ogni oggetto della classe **ListaChar** rappresenta una lista di caratteri implementata mediante una rappresentazione collegata i cui nodi sono oggetti di tipo **NodoChar**. La classe **NodoChar** è la seguente:

```
public class NodoChar {  
  
    private char info;  
    private NodoChar next;  
  
    public NodoChar(char info, NodoChar next){  
        this.info=info;  
        this.next=next;  
    }  
  
    public char getInfo() { return info; }  
  
    public void setInfo(char info) { this.info = info; }  
  
    public NodoChar getNext() { return next; }  
  
    public void setNext(NodoChar next) { this.next = next; }  
}
```

La classe **ListaChar** ha il seguente scheletro:

```
class ListaChar {  
  
    private NodoChar head;  
    private int size;  
  
    /* Crea una lista contenenti i caratteri presenti nella stringa s passata come parametro. I caratteri  
    devono apparire nella lista nello stesso ordine in cui compaiono nella stringa. Se la stringa è  
    vuota o nulla la lista deve essere vuota. */  
    public ListaChar(String s){...}
```

```
/* Riceve in ingresso un array di char caratteri e restituisce un array di interi a della stessa
dimensione di caratteri tale che a[i] è uguale al numero di occorrenze del carattere caratteri[i] nella
lista. Ovviamente, se caratteri[i] non compare nella lista a[i] sarà uguale a 0 */
public int[] contaOccorrenze(char[] caratteri){...}
```

```
/* Confronta dal punto di vista lessicografico la sequenza di caratteri rappresentata dalla lista su cui
viene invocato il metodo e la sequenza di caratteri rappresentata dalla lista l passata come
parametro. Restituisce -1, 0, o 1 a seconda che la lista su cui viene invocato il metodo sia
lessicograficamente precedente, uguale o successiva alla lista l */
public int compareTo(ListaChar l){...}
```

```
/* Restituisce una rappresentazione testuale della lista */
public String toString(){...}
```

```
}
```

A titolo di esempio si immagini che **l1** ed **l2** siano due oggetti di tipo **ListaChar** che rappresentano rispettivamente le sequenze di caratteri **(i,t,a,l,i,a)** e **(i,t,a,l,i,c,o)**. Si supponga inoltre che **c** sia l'array di char **{a,b,c}**. L'invocazione **l1.contaOccorrenze(c)** restituisce l'array di interi **{2,0,0}**; l'invocazione **l1.compareTo(l2)** restituisce **-1** in quanto la sequenza **(i,t,a,l,i,a)** è lessicograficamente precedente alla sequenza **(i,t,a,l,i,c,o)**.

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

- Fa inserire all'utente una stringa **s1**.
- Crea un oggetto **ListaChar l1** corrispondente ad **s1** e lo visualizza.
- Fa inserire all'utente un array di char **c** (dimensione e contenuto scelti dall'utente).
- Stampa all'utente le frequenze dei caratteri di **c** nella lista **l1**.
- Fa inserire all'utente una seconda stringa **s2**.
- Crea un oggetto **ListaChar l2** corrispondente ad **s2** e lo visualizza.
- Dice all'utente se **l1** precede lessicograficamente **l2**, oppure **l2** precede **l1**, oppure sono uguali.

Note:

1. Tutti i file creati devono essere salvati nella cartella C:\fi0625 del proprio PC
2. La cartella C:\fi0625 contiene le classi **InputWindow** e **OutputWindow**.
3. 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.