Cognome _____ Nome _____
Matricola _____ Postazione PC _____

Corso di Laurea in Ingegneria Gestionale
Esame di Informatica - a.a. 2017/2018
23 Luglio 2018

Testo

Il database del sistema di gestione delle tratte autostradali di una compagnia di gestione delle autostrade è costituito da due vettori paralleli. Il primo è denominato “sections” e contiene oggetti di tipo “Tratta” che rappresentano le tratte autostradali presenti all'interno della banca dati della compagnia. Il secondo vettore è denominato “passes” e contiene oggetti di tipo “Passaggio” che rappresentano le informazioni relative ad ogni passaggio effettuato da un veicolo in una determinata tratta autostradale. Ad ogni tratta autostradale può corrispondere più di un passaggio veicolare, in quel caso le informazioni della tratta autostradale saranno replicate.
Per ogni tratta autostradale presente nella posizione i-esima del vettore “sections”, le informazioni relative ad un passaggio veicolare si troveranno nella corrispondente posizione del vettore “passes”. Nel caso in cui la tratta autostradale in posizione i-esima non abbia alcun passaggio veicolare associato, nella posizione corrispondente nel vettore “passess” sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante “MAX_ELEM” (inizializzata a 1024). Se il numero delle tratte contenute nell’archivio è inferiore a “MAX_ELEM”, i primi elementi del vettore conterranno gli oggetti di tipo “Tratta”, mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore “sections” si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Tratta contiene le informazioni relative ad una tratta autostradale:

```
public class Tratta {

	private String codice;	public String ingresso;	public String uscita;
	public double distanza;	public double costo;	public String autostrada;

	public Tratta(String code, String entrance, String exit, double distance, double cost, String highway) {
		codice = code;		ingresso = entrance;		uscita = exit;
		distanza = distance;	costo = cost;		autostrada = highway;
	}

	public String getCodice() {
		return codice;
	}

	public String toString() {
		return	("#" + codice + " [" + autostrada + "]: " + ingresso + " - " + uscita +
				" (Km " + distanza + ", € " + costo + ")");
	}

}
```

La classe Passaggio contiene le informazioni relative ai passaggi veicolari relativi ad una tratta autostradale.

```
public class Passaggio {

	private static int numeroProgressivo = 0;
	private int numero;		public String targa;		public char titoloViaggio;
	public String dataEntrata;	public String dataUscita;

	public Passaggio(String licenseNumber, char ticketType, String entranceDate, String exitDate) {
		targa = licenseNumber;		titoloViaggio = tickeType;
		dataEntrata = entranceDate;	dataUscita = exitDate;
		numero = ++numeroProgressivo;
	}

	public Passaggio(int number, String license, char ticketType, String entranceDate, String exitDate) {
		targa = license;		titoloViaggio = ticketType;
		dataEntrata = entranceDate;	dataUscita = exitDate;
		numero = number;
	}

	public int getNum() {
		return numero;
	}

	public String toString() {
		return ( "[" + numero + "] " + targa + ": " + dataEntrata + " - " + dataUscita +
				" (" + titoloViaggio + ")");
	}

}
```

Si consiglia di procedere implementando un metodo e successivamente la parte del main che utilizza tale metodo. Le varie operazioni devono essere eseguite sulla porzione significativa dell'archivio, cioè la porzione di "sections" che non contiene riferimenti *null*. Se si ha la necessità di convertire una stringa in un *long* oppure in un *int*, si possono utilizzare le funzioni di libreria *Long.parseLong(s)* ed *Integer.parseInt(s)* che convertono la stringa s rispettivamente in un *long* oppure in un *int* restituiti come risultato.

A. Scrivere il metodo statico:

```
public static int contaVeicoli(Tratta[] tratte, Passaggio[] passaggi)
```

Il metodo deve contare il numero di veicoli distinti transitati contromano in tutte le tratte autostradali.

B. Scrivere il metodo statico:

```
public static void ordinaDB(Tratta[] tratte, Passaggio[] passaggi)
```

Il metodo deve ordinare, nell'array parallelo, gli elementi in maniera decrescente usando come criterio la data di uscita di un passaggio. Le tratte senza passaggi associati vanno poste in cima all'array.

C. Scrivere il metodo statico:

```
public static int tempoMedio(Tratta[] tratte, Passaggio[] passaggi,
                                                        String codice)
```

Il metodo deve calcolare il tempo medio di percorrenza, espresso in minuti, di una tratta identificata dal suo codice.

D. Scrivere il metodo statico:

```
public static boolean effettuaPassaggio(Tratta[] tratte, Passaggio[] passaggi,
String codice, String targa, char titoloViaggio, String dataEntrata, String
dataUscita)
```

Il metodo deve inserire nel database specificato dai parametri "tratte" e "passaggi" un nuovo passaggio (individuato dal numero di targa, titolo di viaggio, data di entrata e di uscita) associato ad una tratta che non abbia un passaggio associato oppure duplicando la tratta stessa in caso contrario. Il metodo deve restituire **true** o **false** a seconda del fatto che sia stata trovata almeno una tratta associata al codice passato come parametro nel database. Se non esiste alcuna tratta associata al codice passato come parametro, l'inserimento non deve essere effettuato. L'inserimento deve mantenere l'archivio ordinato come nel punto precedente ed in uno stato consistente.

E. Scrivere il metodo main che:

definisca ed inizializzi i vettori "sections" e "passes" secondo i valori in tabella inizializzando a *null* o a *zero* i campi non presenti. La stampa dell'archivio consiste nello stampare le informazioni di ogni tratta autostradale e dei relativi passaggi veicolari (se ve ne sono). Si utilizzino correttamente i relativi costruttori e metodi *toString()* implementati nelle due classi.

Codice	**Autostrada**	**Ingresso**	**Uscita**	**Distanza** (Km)	**Costo** (€)	**Nr.**	**Targa**	**Data Entrata** (gg/mm/aaaa hh:mm)	**Data Uscita** (gg/mm/aaaa hh:mm)	**Titolo Viaggio**
PNVA12	A12	Pisa Nord	Viareggio	11,7	2,00	2	PG345ZQ	01/05/2018 20:10	01/05/2018 20:15	(T)elepass
GEVA12	A12	Genova Est	Viareggio	131,9	15,00					
FNBCA1	A1	Firenze Nord	Bologna Casalecchio	85,4	8,10	6	QR987LK	11/06/2018 00:05	11/06/2018 01:23	(T)elepass
PNFNA11	A11	Pisa Nord	Firenze Nord	76,4	6,1	5	HJ234FR	01/06/2018 09:36	01/06/2018 09:05	(N)essuno
PNFNA11	A11	Pisa Nord	Firenze Nord	76,4	6,1	7	QR987LK	10/06/2018 22:45	11/06/2018 00:05	(T)elepass
PNVA12	A12	Pisa Nord	Viareggio	11,7	2,00	1	AJ632RR	28/04/2018 07:02	28/04/2018 07:15	(T)elepass
FNBCA1	A1	Firenze Nord	Bologna Casalecchio	85,4	8,10	3	AS768ED	18/05/2018 08:36	18/05/2018 09:45	(B)iglietto
FNBCA1	A1	Firenze Nord	Bologna Casalecchio	85,4	8,10	4	HJ234FR	01/06/2018 09:05	01/06/2018 08:22	(N)essuno

- Avvalendosi del metodo al punto A stampi a video il numero di veicoli distinti transitati contromano in tutte le tratte autostradali.
- Ordini l'intero archivio utilizzando il metodo del punto B e stampi a video l'archivio prima e dopo l'ordinamento.
- Avvalendosi del metodo al punto C stampi a video il tempo medio di percorrenza, espresso in minuti, della tratta Firenze Nord – Bologna Casalecchio.
- Utilizzando il metodo al punto D, effettui l'inserimento del passaggio effettuato dall'autoveicolo targato AJ632RR con Biglietto in data odierna, entrato alle ore 7:30 ed uscito alle ore 8. Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.