

Cognome \_\_\_\_\_ Nome \_\_\_\_\_  
Matricola \_\_\_\_\_ Postazione PC \_\_\_\_\_

Corso di Laurea in Ingegneria Gestionale  
Esame di Informatica - a.a. 2016/2017  
16 Gennaio 2017

### Testo

Il database del sistema di gestione delle tessere “fedeltà” di un supermercato è costituito da due vettori paralleli. Il primo è denominato “cards” e contiene oggetti di tipo “Tessera” che rappresentano le tessere presenti all'interno della banca dati del supermercato. Il secondo vettore è denominato “products” e contiene oggetti di tipo “Prodotto” che rappresentano le informazioni relative ad ogni prodotto acquistato con una determinata tessera. Ad ogni tessera può corrispondere più di un prodotto acquistato, in quel caso le informazioni della tessera saranno replicate.

Per ogni tessera presente nella posizione *i*-esima del vettore “cards”, le informazioni relative ad un prodotto si troveranno nella corrispondente posizione del vettore “products”. Nel caso in cui la tessera in posizione *i*-esima non abbia alcun prodotto associato, nella posizione corrispondente nel vettore “products” sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante “MAX\_ELEM” (inizializzata a 1024). Se il numero di tessere contenute nell'archivio è inferiore a “MAX\_ELEM”, i primi elementi del vettore conterranno gli oggetti di tipo “Tessera”, mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore “cards” si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Tessera contiene le informazioni relative ad un tessera:

```
public class Tessera {  
  
    private static int numeroProgressivo = 0;  
    private int numero;           public String cliente;  
    public String iscrizione;     public String validita;  
  
    public Tessera(String name, String signup, String validity) {  
        numero = numeroProgressivo++; cliente = name;  
        iscrizione = signup;         validita = validity;  
    }  
  
    public Tessera(int number, String name, String signup, String validity) {  
        numero = number;           cliente = name;  
        iscrizione = signup;       validita = validity;  
    }  
  
    public int getNum() {  
        return numero;  
    }  
  
    public String toString() {  
        return ("#" + numero + ": " + cliente + " - " +  
            iscrizione + " ( " + validita + " )");  
    }  
  
}
```

La classe Prodoto contiene le informazioni relative ai singoli prodotti acquistati con una determinata tessera.

```
public class Prodotto {  
  
    private String codice;       public String data;  
    public double prezzo;       public int quantita;  
  
    public Prodotto(String code, String date, double price, int amount) {  
        codice = code;          data = date;  
        prezzo = price;         quantita = amount;  
    }  
  
    public String getCodice() {  
        return codice;  
    }  
  
    public String toString() {  
        return ( "[" + codice + "]" + data +  
            " - " + prezzo + " € (" + quantita + " )");  
    }  
  
}
```

**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 "cards" che non contiene riferimenti "null". Se si ha la necessità di convertire una stringa in intero, si può utilizzare la funzione di libreria *Integer.parseInt(s)* che converte la stringa *s* in un intero restituito come risultato.

**A. Scrivere il metodo statico:**

```
public static int contaTessereAnno(Tessera[] tessere, Prodotto[] prodotti,
                                   int anno)
```

Il metodo deve contare il numero totale di Tessere distinte emesse in un determinato anno, specificato per mezzo del parametro "anno".

**B. Scrivere il metodo statico:**

```
public static void ordinaTessere(Tessera[] tessere, Prodotto[] prodotti)
```

Il metodo deve ordinare, nel vettore "cards", gli elementi in maniera crescente usando come criterio il codice del prodotto acquistato e la relativa data di acquisto. Le tessere con nessun prodotto acquistato vanno in fondo al vettore.

**C. Scrivere il metodo statico:**

```
public static double spesaTotaleMedia(Tessera[] tessere, Prodotto[] prodotti)
```

Il metodo deve calcolare la spesa media di tutti i prodotti acquistati.

**D. Scrivere il metodo statico:**

```
public static boolean inserisciAcquisto(Tessera[] tessere, Prodotto[] prodotti,
                                        int number, String code, String date, double price, int amount)
```

Il metodo deve inserire nel database, specificato dai parametri "tessere" e "prodotti", un acquisto di un prodotto (individuato dal codice, data, prezzo e quantità) associato ad una tessera, individuata per mezzo del numero, che non abbia un acquisto associato oppure duplicando la tessera stessa in caso contrario. Il metodo deve restituire **true** o **false** a seconda del fatto che sia stata trovata almeno una tessera associata al numero passato come parametro nel database. Se non esiste nessuna tessera associata al numero passato come parametro, l'inserimento non deve essere effettuato. L'inserimento deve mantenere l'archivio ordinato come nel punto precedente e in uno stato consistente.

**E. Scrivere il metodo main che:**

definisca ed inizializzi i vettori "cards" e "products" secondo i valori in tabella. La stampa dell'archivio consiste nello stampare le informazioni di ogni tessera e i prodotti acquistati (se ve ne sono). Si utilizzino correttamente i relativi metodi toString() implementati nelle due classi.

Numero	Cliente	Iscrizione (gg/mm/aaaa)	Validità (mm/aaaa)	Codice	Data (gg/mm/aaaa)	Prezzo (€)	Quantità
0	Pietro Rossi	03/02/2013	02/2018	A01	03/02/2013	1,80	2
1	Mario Rossi	15/03/2014	03/2019				
2	Mario Ramarri	02/04/2014	04/2019	A02	02/04/2014	3,50	5
3	Giovanni Verdi	07/07/2016	07/2021	C03	07/07/2016	10,80	10
3	Giovanni Verdi	07/07/2016	07/2021	F18	07/07/2016	4,99	3
0	Pietro Rossi	03/02/2013	02/2018	F18	09/10/2014	4,99	4
4	Mario Ramarri	09/11/2014	11/2019	C03	09/11/2014	10,80	2
2	Mario Ramarri	02/04/2014	04/2019	A01	25/12/2014	1,80	1

- Avvalendosi del metodo al punto A stampi a video, il numero totale delle tessere distinte emesse nel 2014.
- Ordini l'intero archivio utilizzando il metodo del punto B e stampi a video l'archivio prima e dopo l'ordinamento.
- Utilizzando il metodo C, stampi la spesa totale media di tutti i prodotti acquistati.
- Avvalendosi del metodo al punto D, si aggiunga alla tessera n.1 l'acquisto di 3 unità di prodotto "F18" effettuato in data "16/01/2017". Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.