

Cognome _____

Nome _____

Matricola _____

Postazione PC _____

Corso di Laurea in Ingegneria Gestionale

Esame di Informatica - a.a. 2015/2016

25 Luglio 2016

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 contaClienti(Tessera[] tessere)
```

Il metodo deve contare il numero totale dei clienti che abbiano sottoscritto almeno una tessera. Nel caso in cui un cliente abbia sottoscritto più di una tessera, esso deve essere conteggiato una sola volta.

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 prima il cognome e poi il nome del cliente associato alla tessera. Nel caso si abbiano clienti omonimi, come ulteriore criterio dovrà essere utilizzato il numero di prodotti acquistati in maniera decrescente.

C. Scrivere il metodo statico:

```
public static int puntiFedelta(Tessera[] tessere, Prodotto[] prodotti,  
                               int numero)
```

Supponendo che il supermercato assegni dei punti fedeltà ad ogni tessera assegnando 1 punto per ogni prodotto acquistato, raddoppiando i punti di ogni acquisto se la quantità di prodotto supera le 3 unità e, triplicando i punti di ogni acquisto se la quantità di prodotto supera le 5 unità; il metodo deve calcolare il totale dei punti accumulati da una determinata tessera individuata dal suo numero.

D. Scrivere il metodo statico:

```
public static boolean rimuoviDuplicati(Tessera[] tessere,  
                                       Prodotto[] prodotti, int numero)
```

Il metodo deve eliminare dal database, specificato dai parametri "tessere" e "prodotti", tutte le tessere duplicate di una determinata tessera individuata dal suo numero; restituire **true** o **false** a seconda del fatto che sia stato eliminata almeno una tessera e mantenere l'archivio 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	1
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 dei clienti che hanno sottoscritto almeno una tessera.
- 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 il totale dei punti della tessera numero 2.
- Rimuova i duplicati della tessera numero 2 utilizzando il metodo del punto D. Al termine dell'operazione si stampi l'archivio aggiornato se l'operazione è avvenuta con successo, altrimenti si stampi un messaggio di errore.