Esercizio 1. Si deve modellare, tramite una classe Java, una squadra di calcio. Gli elementi di base che costituiscono una squadra sono i giocatori, caratterizzati da un nome, un ruolo ed il valore di mercato (espresso tramite un valore intero). La squadra comprende un insieme di 26 giocatori tesserati (chiamato 'rosa') ed una formazione di 11 giocatori titolari che scenderanno in campo per la prossima partita. La formazione deve essere composta secondo uno schema 4-4-2 e cioè: innanzitutto il portiere, poi 4 difensori, 4 centrocampisti e 2 attaccanti. I giocatori della formazione devono essere necessariamente giocatori appartenenti alla rosa di giocatori in squadra.

Si implementino la classe Giocatore con i metodi che si ritiene opportuno definire, e la classe Squadra con almeno i seguenti metodi (se necessario, si potranno ovviamente definire altri metodi ausiliari):

- **costruttore privato** (utilizzato dal metodo "leggi()") : riceve in input la rosa di giocatori della squadra.
- **leggi**: metodo statico per la lettura da input di un oggetto squadra. Devono essere letti da input i 26 giocatori della rosa, ad ognuno dei quali viene assegnato un numero di maglia che va da 1 a 26.
- **schiera**: metodo che determina la formazione. Devono essere letti da input i nomi degli 11 giocatori da schierare nella prossima partita, nell'ordine: il portiere, i 4 difensori, i 4 centrocampisti, i 2 attaccanti. Il metodo deve verificare che i nomi inseriti corrispondano a giocatori effettivamente presenti nella rosa e che abbiano il ruolo con cui vengono schierati. **Nota:** Si dovrebbe anche controllare che uno stesso giocatore non venga inserito più di una volta nella formazione, ma questo controllo è opzionale: se svolto correttamente, verrà riconosciuto un bonus.
- **getMagliaGiocatore**: metodo che riceve in input un nome ed un ruolo e restituisce il numero di maglia del giocatore con quel nome e quel ruolo, presente nella rosa. Il metodo deve restituire -1 se nella rosa non esiste nessun giocatore con quelle caratteristiche
- **getNomeGiocatore**: metodo che riceve in input un numero di maglia e restituisce il nome del giocatore della rosa a cui è stato assegnato quel numero di maglia.
- **stampaRosa**: metodo che stampa in output tutti i giocatori della rosa, raggruppati per ruolo: prima tutti i portieri, poi (su righe diverse) tutti i difensori, tutti i centrocampisti e infine tutti gli attaccanti.
- **stampaFormazione**: metodo che stampa in output tutti i giocatori della formazione che si è deciso di schierare: il portiere, i 4 difensori, i 4 centrocampisti e infine i 2 attaccanti.
- **stampaValore**: metodo che stampa in output il valore complessivo dei giocatori della squadra (cioè la somma dei valori di mercato di tutti i giocatori nella rosa).

SOLUZIONE:

Classe GIOCATORE:

```
import java.util.*;

public class Giocatore {

	public enum Ruoli {portiere, difensore, centrocampista, attaccante};

	private String nome;
	private Ruoli ruolo;
	private int valore;

	public Giocatore(){
		nome = "";
		ruolo = Ruoli.portiere;
		valore = 0;
	}

	private Giocatore(String n, Ruoli r, int v){
		nome = n;
		ruolo = r;
		valore = v;
	}

	public Giocatore(Giocatore g){
		nome = g.nome;
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

```
                ruolo = g.ruolo;
                valore = g.valore;
        }

        public static Giocatore leggi(Scanner input){
        System.out.println("Nome: ");
        String n = input.next();
         System.out.println("Ruolo (portiere | difensore | centrocampista | attaccante ): ");
        String r = input.next();
        System.out.println("Valore: ");
        int v = input.nextInt();
        Giocatore g = new Giocatore(n, Ruoli.valueOf(r), v);
        return g;
        }

        public String getNome(){
                return nome;
        }

        public Ruoli getRuolo(){
                return ruolo;
        }

        public int getValore(){
                return valore;
        }

        public String toString(){
                return nome + " ";
        }
}
```

Classe SQUADRA:

```
import java.util.*;

public class Squadra {

        private Giocatore[] rosa = new Giocatore[26];
        private Giocatore[] formazione = new Giocatore[11];

        private Squadra(Giocatore[] r){
                rosa = r;
        }
        public static Squadra leggi(Scanner input){
        System.out.println("Giocatori della squadra: ");
        Giocatore[] r = new Giocatore[26];
        for( int i=0; i<r.length; i++ ){
                System.out.println("maglia num " + (i+1) + ":");
                r[i] = Giocatore.leggi(input);
        }
        Squadra s = new Squadra(r);
        return s;
        }

        public int getMagliaGiocatore(String g, Giocatore.Ruoli r){
        for( int i=0; i<rosa.length; i++ ){
                if( rosa[i].getNome().equals(g) && rosa[i].getRuolo()==r)
                        return i+1;
        }
        return -1;
        }

        public String getNomeGiocatore(int maglia){
        return rosa[maglia].getNome();
        }

        public void schiera(Scanner input){
        System.out.println();
        System.out.println("Giocatori da schierare: ");


        System.out.println("portiere?");
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

```
        String g = input.next();
        int maglia = getMagliaGiocatore(g, Giocatore.Ruoli.portiere);
        if( maglia < 0 ){
                System.out.println("Errore");
                return;
        }
        formazione[0] = rosa[maglia-1];

        System.out.println("difensori?");
        for(int i=1; i<5; i++){
                g = input.next();
                maglia = getMagliaGiocatore(g, Giocatore.Ruoli.difensore);
                if( maglia < 0 ){
                        System.out.println("Errore");
                        return;
                }
                formazione[i] = rosa[maglia-1];
        }

        System.out.println("centrocampisti?");
        for(int i=5; i<9; i++){
                g = input.next();
                maglia = getMagliaGiocatore(g, Giocatore.Ruoli.centrocampista);
                if( maglia < 0 ){
                        System.out.println("Errore");
                        return;
                }
                formazione[i] = rosa[maglia-1];
        }

        System.out.println("attaccanti?");
        for(int i=9; i<11; i++){
                g = input.next();
                maglia = getMagliaGiocatore(g, Giocatore.Ruoli.attaccante);
                if( maglia < 0 ){
                        System.out.println("Errore");
                        return;
                }
                formazione[i] = rosa[maglia-1];
        }
        }

        public void stampaRosa(){
        System.out.println();
        System.out.println("Portieri: ");
                for(int i = 0; i < rosa.length; i++)
                        if(rosa[i].getRuolo() == Giocatore.Ruoli.portiere)
                                System.out.print(rosa[i]);
        System.out.println();
        System.out.println("Difensori: ");
                for(int i = 0; i < rosa.length; i++)
                        if(rosa[i].getRuolo() == Giocatore.Ruoli.difensore)
                                System.out.print(rosa[i]);
        System.out.println();
                System.out.println("Centrocampisti: ");
                for(int i = 0; i < rosa.length; i++)
                        if(rosa[i].getRuolo() == Giocatore.Ruoli.centrocampista)
                                System.out.print(rosa[i]);
        System.out.println();
                System.out.println("Attaccanti: ");
                for(int i = 0; i < rosa.length; i++)
                        if(rosa[i].getRuolo() == Giocatore.Ruoli.attaccante)
                                System.out.print(rosa[i]);
        }

        public void stampaFormazione(){
        System.out.println();
        System.out.println("Formazione in gioco:");
        System.out.println("Portiere: ");
                System.out.print(formazione[0]);
        System.out.println();
        System.out.println("Difensori: ");
                for(int i = 1; i < 5; i++)
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

```
                        System.out.print(formazione[i]);
        System.out.println();
                System.out.println("Centrocampisti: ");
                for(int i = 5; i < 9; i++)
                        System.out.print(formazione[i]);
        System.out.println();
                System.out.println("Attaccanti: ");
                for(int i = 9; i < 11; i++)
                        System.out.print(formazione[i]);
        }

        public void stampaValore(){
        System.out.println();
        System.out.println("Valore della squadra:");
        int valore = 0;
                for(int i = 0; i < rosa.length; i++)
                        valore += rosa[i].getValore();
                System.out.print(valore);
        }
}
```

Esercizio 2. Sia A l'insieme ricorsivo di numeri naturali definito come segue:

$$\begin{cases} 0 \notin A \\ 1 \in A \\ x \in A \iff \dfrac{2x-1}{4} \in A, \quad \forall \quad x > 1 \end{cases}$$

Si implementi un metodo statico JAVA che, ricevuto un intero x, restituisca ***true*** se x appartiene ad A, **false** altrimenti.
NOTA: La divisione si intende fra interi (non negativi), quindi da approssimare per difetto.

SOLUZIONE:

```
    public static boolean A(int x) {
       if(x == 0)
           return false;
       if(x == 1)
           return true;
       return A((2*x-1)/4);
    }
```

Esercizio 3. Considerare il seguente frammento di codice:

```
    void f(int[] v) {
        int n = v.length;
        for(int i = 0; i < n/2; i++) {
            int tmp = v[i];
            v[i] = v[n-i];
            v[n-i] = tmp;
        }
    }
    int[] v = {1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1};
```

Indicare:

a) quale sarà il contenuto del vettore v in seguito alle seguenti istruzioni:

CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it

```
int[] v = {1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1};
f(v);
```

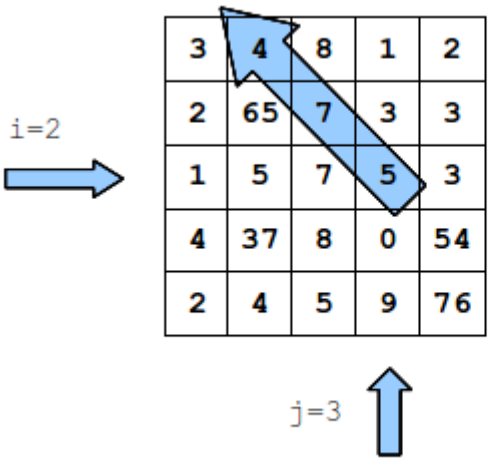
b) una breve descrizione (max 50 parole) del metodo, ovvero il modo in cui modifica il vettore v.

SOLUZIONE:

1. `{1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1}`

2. Il metodo f() riceve un vettore di interi v e inverte la posizione dei suoi elementi: il primo viene scambiato con l'ultimo, il secondo con il penultimo, ...

Esercizio 4. Implementare un metodo che, data una matrice rettangolare M di interi e le coordinate i,j di una cella, restituisca la somma degli elementi "posti diagonalmente a nord-ovest" rispetto a i,j (cella i,j inclusa).
ESEMPIO:



RISULTATO: 5+7+4=16

SOLUZIONE:

```
public static int g(int[][] m, int i, int j) {
    int sum = m[i][j];
    while(i > 0 && j > 0) {
        i--; j--;
        sum += m[i][j];
    }
    return sum;
}
```


CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410
http://sv.mat.unical.it