Esercizio 1. Si deve modellare, tramite una classe Java, un mobile libreria con tre scaffali. Per semplicità, si suppone che ogni scaffale possa contenere al più venti libri. Un libro è caratterizzato da un titolo, rappresentato come una stringa di caratteri che si suppone non contenga spazi bianchi.
Si implementino almeno i seguenti metodi (se necessario, si potranno ovviamente definire altri metodi ausiliari):

- **costruttore privato** (utilizzato dal metodo "leggi()") : riceve in input il contenuto dei tre scaffali.
- **costruttore per copia**: riceve in input una libreria.
- **leggi**: metodo statico per la lettura da input di un oggetto libreria. Si consideri la sequenza di libri da leggere da input, terminata dalla stringa "fine", che funge da tappo.
- **getScaffale**: metodo che riceve in input un numero di scaffale (1, 2, o 3) e restituisce i libri contenuti in quello scaffale.
- **svuotaScaffale**: metodo che riceve in input un numero di scaffale e svuota lo scaffale specificato.
- **svuotaLibreria**: metodo che svuota tutti gli scaffali della libreria.
- **getLibro**: metodo che riceve in input un numero di scaffale ed una posizione all'interno di quello scaffale e restituisce il libro contenuto in quella posizione (restituisce la stringa vuota nel caso in cui quella posizione sia libera).
- **cercaLibro**: metodo che riceve in input un libro (il suo titolo) e, se il libro si trova nella libreria, restituisce 'true' e stampa il numero di scaffale e la posizione in cui esso si trova, altrimenti restituisce 'false' senza produrre alcuna stampa.
- **aggiungiLibro**: metodo che riceve in input un libro (il suo titolo) e lo inserisce nel primo posto disponibile della libreria; se l'inserimento ha successo il metodo restituisce 'true', altrimenti (la libreria non ha posizioni disponibili) restituisce 'false'.
- **compattaScaffale**: metodo che riceve in input un numero di scaffale e ha come effetto la compattazione dei libri contenuti in quello scaffale. Uno scaffale con N libri è compattato se tutti gli N libri sono collocati nelle prime N posizioni dello scaffale.
- **compattaLibreria**: metodo che ha come effetto la compattazione dei libri per tutta la libreria. Una libreria è compattata se non esiste nessuna posizione libera tra due libri, e i libri occupano le posizioni a partire dalle prime posizioni del primo scaffale. **Nota:** l'implementazione di tale metodo è opzionale, se svolto correttamente verrà riconosciuto un bonus.

SOLUZIONE:

```
public class Scaffale {

        private String[] libri = new String[20];

        public Scaffale(){
                for( int i=0; i<libri.length; i++)
                        libri[i]="";
        }

        private Scaffale(String[] l){
                libri = l;
        }

        public Scaffale(Scaffale s){
                for( int i=0; i<libri.length; i++)
                        libri[i]= s.libri[i];
        }

        public static Scaffale leggi(Scanner input){
                System.out.println("libri dello scaffale, 'fine' per terminare (max 20): ");
                String[] l = new String[20];
                int i = 0;
                String libro = input.next();
                while( !libro.equals("fine") && i<l.length ){
```


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

```
                l[i] = libro;
                libro = input.next();
                i++;
        }
        for( ; i<l.length; i++)
                l[i]="";
        Scaffale s = new Scaffale(l);
        return s;
}

public void svuota(){
        for( int i=0; i<libri.length; i++)
                libri[i]="";
}

public String getLibro(int p){
        return libri[p-1];
}

public boolean cercaLibro(String l){
        for( int i=0; i<libri.length; i++)
                if(libri[i].equals(l)){
                        System.out.println("Il libro si trova in posizione " + i+1);
                        return true;
                }
        return false;
}

public boolean aggiungiLibro(String l){
        for( int i=0; i<libri.length; i++)
                if(libri[i].equals("")){
                        libri[i] = l;
                        return true;
                }
        return false;
}

public boolean inserisciLibro(String l, int p){
        if(libri[p-1].equals("")){
                libri[p-1] = l;
                return true;
        }
        return false;
}

public void compatta(){
        boolean compatto = false;
        for(int i=0; i<libri.length && !compatto; i++)
                if(libri[i].equals("")){
                        int j=i+1;
                        boolean trovato = false;
                        while(j<libri.length && !trovato){
                                trovato = !libri[j].equals("");
                                j++;
                        }
                        if(trovato){
                                libri[i] = libri[j-1];
                                libri[j-1] = "";
                        }
                        else
                                compatto = true;
                }
}
```


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

```
	public String toString(){
		String output = "";
		for(int i = 0; i < libri.length; i++)
			output += libri[i] + " ";
		return output;
	}

	public int size(){
		return libri.length;
	}

}

public class Libreria {

	private Scaffale[] scaffali = new Scaffale[3];

	private Libreria(Scaffale[] s){
		scaffali = s;
	}

	public Libreria(Libreria l){
		for( int i=0; i<scaffali.length; i++)
			scaffali[i] = l.scaffali[i];
	}

	public static Libreria leggi(Scanner input){
		System.out.println("Scaffali della libreria: ");
		Scaffale[] s = new Scaffale[3];
		for( int i=0; i<s.length; i++ ){
			System.out.println("scaffale " + (i+1) + ":");
			s[i] = Scaffale.leggi(input);
		}
		Libreria l = new Libreria(s);
		return l;
	}

	public Scaffale getScaffale(int s){
		Scaffale sc = new Scaffale(scaffali[s-1]);
		return sc;
	}

	public void svuotaScaffale(int s){
		scaffali[s-1].svuota();
	}

	public void svuotaLibreria(){
		for( int i=0; i<scaffali.length; i++)
			scaffali[i].svuota();
	}

	public String getLibro(int s, int p){
		return scaffali[s-1].getLibro(p);
	}

	public boolean cercaLibro(String l){
		for( int i=0; i<scaffali.length; i++)
			if(scaffali[i].cercaLibro(l)){
			System.out.println("Il libro si trova nello scaffale " + i+1);
				return true;
			}
		return false;
```


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

```
        }

        public boolean aggiungiLibro(String l){
                for( int i=0; i<scaffali.length; i++)
                        if(scaffali[i].aggiungiLibro(l))
                                return true;
                return false;
        }

        public boolean inserisciLibro(String l, int s, int p){
                if(scaffali[s-1].inserisciLibro(l, p))
                        return true;
                return false;
        }

        public void compattaScaffale(int s){
                scaffali[s-1].compatta();
        }

        public void compattaLibreria(){
                for(int i = 1; i <= scaffali.length; i++){
                        Scaffale s = getScaffale(i);
                        svuotaScaffale(i);
                        for(int j = 1; j <= s.size(); j++)
                                if(!s.getLibro(j).equals(""))
                                        aggiungiLibro(s.getLibro(j));
                }
        }

        public String toString(){
                String output = "";
                for(int i = 0; i < scaffali.length; i++)
                        output += scaffali[i] + "\n";
                return output;
        }
}
```

Esercizio 2. E` data la seguente classe Java.

```
public static int f(int x) {
    if(x > 0) {
        int y = f(x – 1);
        return y + 1;
    }
    return x + 1;
}
```

```
public static int g(int[] v) {
    int b = 0;
    int c = 0;
    while(true) {
        if(c >= v.length)
            return b;
        b += v[c];
        c++;
    }
}
```

```
public static void main() {
    int[] v = new int[10];
    for(int i = 0; i < v.length; i++)
        v[i] = f(i);
    System.out.println(g(v));
}
```

Si svolgano i seguenti punti.

a) Descrivere il comportamento del metodo "f()" (indipendentemente da come viene utilizzato, ovvero descrivere input e output).
b) Descrivere il comportamento del metodo "g()" (indipendentemente da come viene utilizzato, ovvero descrivere input e output).
c) Descrivere il comportamento del metodo "main()", indicando cosa verrà stampato.

SOLUZIONE:

a) Il metodo "f(x)" riceve in input un intero "x" e restituisce "x+1". E' un metodo ricorsivo.
b) Il metodo "g(v)" riceve in input un vettore "v" e restituisce la somma degli elementi di "v".

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

c) Il "main()" inizializza un vettore "v" di 10 elementi assegnando f(i) = i+1 all'elemento i-esimo. L'invocazione di g(v), quindi, restituisce 1+2+...+10 = (10*11)/2 = 55. Pertanto, il "main()" stamperà 55.

Esercizio 3. Si implementi in Java il metodo statico **"verificaArray"** che effettua i controlli, specificati di seguito, su di un array di caratteri "v". In particolare, il metodo dovrà restituire '**true**' se:

- ad ogni carattere non numerico (tutti i caratteri tranne '**0**'..'**9**') presente nella prima metà dell'array corrisponde, nella seconda metà dell'array, in posizione analoga, esattamente lo stesso carattere;
- ad ogni carattere numerico ('**0**'..'**9**') presente nella prima metà dell'array corrisponde, anche nella seconda metà dell'array, in posizione analoga, un carattere numerico (non necessariamente lo stesso).

Il metodo dovrà restituire '**false**' in tutti gli altri casi.
Per esempio, per

v = ['x', 'z', '4', '9', 'y', '8', 'x', 'z', '5', '2', 'y', '3']

il metodo dovrebbe restituire '**true**', mentre per

v = ['w, 'z', '4', '9', 'y', '8', 'x', 'z', '5', '2']

dovrebbe restituire '**false**'.
NOTA1: Si può assumere che il vettore "v" abbia come dimensione un numero pari di elementi.
NOTA2: verrà riconosciuto un bonus se il metodo implementato è ricorsivo.

SOLUZIONE:

```
public static boolean èCifra(char c)
{
        if (c >= '0' && c <= '9')
                return true;
        return false;
}

public static boolean verificaArray(char v[], int corr) {
   if ( corr == v.length/2 )
       return true;
   int sec = v.length/2 +corr;
   if ( ( èCifra(v[corr]) && !èCifra(v[sec]) ) ||
        (!èCifra(v[corr]) && v[corr]!= v[sec] ) )
        return false;
   else
        return verificaArray(v, corr+1);
}
```

Esercizio 4. Si implementi in Java un metodo avente il seguente prototipo:

public static void ruotaDi90GradiAntiorario (**int** m[][])

La matrice (array bidimensionale) di numeri interi "m" si suppone quadrata e già inizializzata. Il metodo deve ruotare "m" di 90 gradi in senso antiorario.

8	7	4	3
5	8	1	3
2	6	9	1
1	2	4	5

ESEMPIO: Se la matrice "m" fosse quella riportata qui di fianco, il metodo "ruotaDi90GradiAntiorario()" dovrebbe modificare "m" nel seguente modo:

3	3	1	5
4	1	9	4
7	8	6	2
8	5	2	1

BONUS: Implementare un metodo avente il seguente prototipo:

public static void ruotaAntiorario (**int** m[][], **int** gradi)

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

Il parametro "gradi" indica di quanti gradi deve essere ruotata la matrice (assumiamo sia un numero nell'intervallo [0,999], e che sia divisibile per 90).
Il metodo ruotaAntiorario() dovrebbe fare uso del metodo ruotaDi90GradiAntiorario() precedentemente implementato.

SOLUZIONE:

```
public static void ruotaDi90GradiAntiorario(int[][] m) {
        int[][] n = copiaMatrice(m);

        for(int i = 0; i < m.length; i++)
            for(int j = 0; j < m.length; j++)
                m[m.length-j-1][i] = n[i][j];
}

public static int[][] copiaMatrice(int[][] m) {
        int[][] n = new int[m.length][m.length];

        for(int i = 0; i < m.length; i++)
            for(int j = 0; j < m.length; j++)
                n[i][j] = m[i][j];

        return n;
}
```

BONUS:

```
public static void ruotaAntiorario(int[][] m, int gradi) {
        gradi %= 360;

        switch (gradi) {
        case 90:
          ruotaDi90GradiAntiorario(m);
          break;

        case 180:
          ruotaDi90GradiAntiorario(m);
          ruotaDi90GradiAntiorario(m);
          break;

        case 270:
          ruotaDi90GradiAntiorario(m);
          ruotaDi90GradiAntiorario(m);
          ruotaDi90GradiAntiorario(m);
          break;

        default:
          break;
        }
}
```

OPPURE:

```
public static void ruotaAntiorario(int[][] m, int gradi) {
  if (gradi > 0) {
    ruotaDi90GradiAntiorario(m);
    ruotaAntiorario(m, gradi - 90);
  }
}
```


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