

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

## Programmazione 1 A.A. 2015/2016

### Appello del 7 febbraio 2017

#### Esercizio 1 (6 punti)

1.1 (3 punti) Cosa stampa il seguente frammento di codice Java?

```
boolean trovato=false;
int[] a = {4, 6, 8, -3, 10};
int n=16, i=-10, j=-10;
for (i=0; i<a.length && !trovato; i++) {
    for (j=i+1; j<a.length && !trovato; j++) {
        if (a[i]+a[j]==n){
            trovato=true;
            System.out.println(i + " e " + j);
        }
    }
}
System.out.println ("i = " + i);
System.out.println ("j = " + j);
```

1.2 (3 punti) Cosa stampa il seguente programma Java?

```
public class Main {
    public static void main(String[] args) {
        System.out.println(enigma(6,2));
        System.out.println(enigma(10,1));
        System.out.println(enigma(1000,4));
    }
    static int enigma (int x, int y){
        if (x<=0 || y<=0) return 0;
        return 1+enigma(x-y,y);
    }
}
```

#### Esercizio 2 (10 punti)

Scrivere un metodo

**static int[] espandi (int[] elementi, int[] molteplicita)**

che, presi come parametro due array di numeri interi *della stessa lunghezza n*, crea e restituisce un array che contenga, per ogni  $i=0, \dots, n-1$ ,  $molteplicita[i]$  copie consecutive dell'elemento  $elementi[i]$  (se  $molteplicita[i]$  non è positivo  $elementi[i]$  non viene inserito nell'array). Se **elementi** o **molteplicita** vale **null**, oppure se non hanno la stessa lunghezza, viene restituito **null**.

Ad esempio se **elementi**={1, 3, -4, 5, 6}, e **molteplicita**={2, 1, 3, 0, -5}, viene creato e restituito l'array {1, 1, 3, -4, -4, -4}.

---

### Esercizio 3 (10 punti)

Si consideri il tipo di dato

```
class Appello {
    int[] matricole;
    int[] voti;
}
```

che rappresenta un appello di esame, in cui **matricole e voti** sono due array della stessa lunghezza e “paralleli”, ovvero tali che, per ogni posizione  $i$  (tra 0 e  $voti.length-1$ ), nella posizione  $i$  di voti è presente il voto riportato dallo studente presente nella stessa posizione  $i$  di matricole.

- **(5 punti)** Scrivere un metodo **static int contoSufficienti (Appello a)** che, presi come parametri un un Appello **a**, restituisce il numero di studenti che hanno conseguito un voto almeno uguale a 18. Se **a** vale null, viene restituito 0.
- **(5 punti)** Scrivere un metodo **static int[] estraiSufficienti (Appello a)** che, facendo uso del metodo contoSufficienti, preso come parametro un Appello **a**, crea e restituisce un array contenente tutte le matricole che hanno riportato un voto sufficiente all'appello. Se **a** vale null viene restituito null.

---

### Esercizio 4 (6 punti)

Si consideri il seguente tipo di dati visto a lezione, che rappresenta una lista di numeri interi, **identificata dal suo primo elemento**.

```
class Elem {
    int valore;
    Elem next;
}
```

Scrivere un metodo iterativo

**public static int conteggio (Elem lista)**

che presa come parametro una lista di interi, restituisce il numero di elementi seguiti da due elementi uguali. Se ad esempio la lista è  $1 \rightarrow 3 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 5 \rightarrow 5 \rightarrow 6$ , il metodo deve restituire 1. Se la lista è vuota (cioè vale null), viene restituito 0.

#### Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **90** minuti.
- Scrivere **subito** nome, cognome, matricola su **OGNI FOGLIO**.
- Le risposte al primo esercizio devono essere date direttamente nei riquadri di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato e depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.