Cognome e Nome _____

Matricola _____

Programmazione e Algoritmi I
A.A. 2018/2019

Appello del 9 settembre 2019

Esercizio 1 (6 punti)

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

```
int conto=0;
int[] a = {-3, -6, 5, 10, -4};
int n=-30, i, j;
for (i=a.length-1; i>=0; i--) {
    for (j=i-1; j>=0; j--) {
        if (a[i]*a[j]==n){
          conto++;
          System.out.println(i + " e " + j);
        }
     }
}
System.out.println ("conto = " + conto);
```

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

```
public class Main {
   public static void main(String[] args) {
        System.out.println(enigma(8,1));
        System.out.println(enigma(7,3));
        System.out.println(enigma(500,4));
   }
    static int enigma (int x, int y){
        if (x<=0 || y<=0) return 0;
        return 2+enigma(x-y,y);
    }
}
```

Esercizio 2 (10 punti)

Scrivere un metodo in Java

`static void ordinamentoRelativo (int[] a, int rif)`

che, preso come parametro un array **a** di numeri interi e un intero **rif**, ordina gli elementi di **a** in modo non decrescente rispetto alla loro distanza (in valore assoluto) dall'intero **rif**.

Ad esempio, se **a**={8, 1, 3, 1, 5, 10, 1, 3} e **rif=**4, a dovrà essere ordinato così: {3, 5, 3, 1, 1, 1, 8, 10}.

Si analizzi la complessità temporale del metodo proposto: tale metodo deve avere complessità temporale **O(n log n)** nel caso peggiore (*soluzioni con complessità temporale peggiore danno luogo a una valutazione minore, pari a un massimo di 6 punti totali*), dove n è la lunghezza dell'array **a** in input.

Esercizio 3 (10 punti)

Si consideri il seguente array di numeri interi:

{20, 2, 35, 1, 3, 18, 60, 55, 41, 22, 19, 5}

Mostrare **passo-passo l'esecuzione dell'heap-sort** per ordinare l'array in modo non decrescente (compresa la fase iniziale di build-max-heap).

Esercizio 4 (9 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.

- **(4 punti)** Scrivere un metodo
  `static int contoIntervallo(Appello a, int min, int max)`
  che, presi come parametri un un Appello **a** e due interi **min** e **max**, restituisce il numero di studenti che hanno conseguito un voto compreso tra **min** e **max** (estremi inclusi). Il metodo deve gestire opportunamente tutti i casi in cui le variabili di tipo riferimento coinvolte valgano *null*.
- **(5 punti)** Scrivere un metodo
  `static int[] escludiIntervallo(Appello a, int min, int max)`
  che, facendo uso del metodo contoIntervallo, preso come parametro un Appello **a** e due interi **min** e **max**, crea e restituisce un array contenente tutte e sole le matricole che **non** hanno riportato un voto compreso (estremi inclusi) tra **min** e **max**. Il metodo deve gestire opportunamente tutti i casi in cui le variabili di tipo riferimento coinvolte valgano *null*.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **100** minuti.
- Scrivere **subito** nome, cognome, matricola su OGNI FOGLIO (**compreso questo**).
- Le risposte al primo esercizio devono essere date direttamente nel riquadro 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
- È possibile consultare appunti, libri e dispense.
- 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.