

**Appello di Programmazione e Algoritmi 1**  
**13 giugno 2022 in presenza**

**Esercizio 1**

Cosa stampa il seguente programma Java?

```
public class provaEsame {
    public static void main(String[] args) {
        System.out.println(prova(6));
        System.out.println(prova(4));
        System.out.println(prova(1));
    }
    static int prova (int x){
        if (x<=0 || x>=10) return x;
        if (x>=5) return -x-prova(x+2);
        return x+ prova (x-2);
    }
}
```

**NON SARANNO CONSIDERATE VALIDE LE RISPOSTE NON GIUSTIFICATE DAI CALCOLI EFFETTUATI PER OTTENERE IL RISULTATO**

Cosa stampa il seguente frammento di codice Java?

```
int k=3, c = 0;
boolean p = false;
for(int i=1;i < a.length && !p; i++) {
    if (a[i-1] >= a[i])
        {c++; System.out.println(a[i-1]); if (c > k) p = true;}
    i++;
}
System.out.println(c); System.out.println(p);
```

**NON SARANNO CONSIDERATE VALIDE LE RISPOSTE NON GIUSTIFICATE DAI CALCOLI EFFETTUATI PER OTTENERE IL RISULTATI**

**Esercizio 2**

Scrivere un metodo in Java

**static int spostaASinistra (int[] a)**

che preso in input un array **a** di elementi interi sposta il contenuto dell'array a di una posizione verso sinistra, inserendo uno 0 nell'ultima posizione e restituisce l'elemento nella prima posizione dell'array originario.

Ad esempio: se **a** = {7,8,-3,42,5}, dopo l'invocazione del metodo **spostaASinistra** con input **a**, il contenuto di **a** diventerà {8,-3,42,5,0} e il valore restituito dal metodo sarà 7. Nel caso in cui l'array sia null o non contenga elementi, il metodo lascerà invariato l'array a e restituirà il valore 0.

**Si analizzi la complessità temporale del metodo proposto** (soluzioni con complessità temporale peggiore danno luogo a una valutazione minore).

### Esercizio 3

Scrivere un metodo in Java

```
static int[] termina(int[] a, int n)
```

che preso in input un array **a** di elementi interi, un intero **n**, restituisce un nuovo array ottenuto da **a** inserendo alla fine di ogni sequenza (non vuota) di occorrenze consecutive di **n** (nell'array **a**) il valore 0.

Ad esempio: se **a**={5, 7, -9, 5, 5, 5, -3}, **n** è uguale a 5, allora viene restituito l'array {5, 0, 7, -9, 5, 5, 5, 0, -3} (viene messo un nuovo elemento di valore 0 alla fine di ogni sequenza (non vuota) di valori uguali a 5).

Altro esempio, se **a**={-7, 1, 3, 1, 1,1}, **n** è uguale a 1, il metodo dovrà restituire {-7, 1, 0, 3, 1, 1,1, 0}.

Se **n** non compare in **a**, allora il metodo dovrà restituire un nuovo array (possibilmente anche null) che contiene esattamente gli stessi elementi di **a**.

**Si analizzi la complessità temporale del metodo proposto** (*soluzioni con complessità temporale peggiore danno luogo a una valutazione minore*).

### Esercizio 4

Si consideri il seguente array di numeri interi:

```
{27, 16, 26, 39,12, 24, 10, 20, 18, 31, 22}
```

Mostrare **passo-passo l'esecuzione del merge-sort** per ordinare l'array in modo non decrescente.

Si descriva la complessità computazionale del merge-sort.

**[Esercizio di Laboratorio]** SOLO PER GLI STUDENTI CLEII che hanno svolto il progetto a partire dall'AA 2020/2021 In relazione al progetto di laboratorio,

Si sviluppi un metodo Java:

```
public static String stringaInversa(int[] a)
```

che restituisce una stringa composta dai valori dell'array **a** in ordine inverso. Ad esempio, dato l'array {1,3,-2,8,0,5} restituisce la stringa "[5,0,8,-2,3,1]".

Si sviluppi un metodo Java:

```
public static int[] posizioniPari(int[] a)
```

che restituisce un nuovo array ottenuto selezionando gli elementi dell'array **a** in posizione pari. Ad esempio, dato l'array {1,3,5,-2,0,9} restituisce l'array {1,5,0}.