

Appello di Programmazione e Algoritmi 1 6 luglio 2022

Esercizio 1

Cosa stampa il seguente programma Java?

```
public class provaEsame {
    public static void main(String[] args) {
        System.out.println(rebus(12));
        System.out.println(rebus(11));
        System.out.println(rebus(7));
    }
    static int rebus (int x){
        if (x<=5) return -x;
        if (x<=10) return x- rebus(x-4);
        return x+rebus(x-10);
    }
}
```

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

Cosa stampa il seguente frammento di codice Java?

```
int[] b = {6,1,7,1,1,4,6};
boolean diversi=true; int i=0;
while(i<b.length-1 && diversi)
    if(b[i]==b[i+1]) diversi=false;
    else { System.out.println("somma: "+ (b[i]+b[i+1])); i++;}
System.out.println("inidce i: " + i);
```

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

Esercizio 2

Scrivere un metodo in Java

static void stampaCoppiePositive (int[] a, int k)

che preso in input un array **a** di elementi interi ed un intero **k**, stampa le prime **k** coppie di numeri positivi presenti in **a**. Se ci sono meno di **k** coppie di numeri positivi stampa tutte quelle presenti.

Ad esempio: se **a** = {-5, 6, -11, -3, -9, 15, -1, 90, -21, 3} e **k** =4, il metodo stamperà 6 15, poi 6 90, poi 6 3 e infine 15 90. Se invece **a** = {-5, 6, -11, -3, -9, 15, -1, 90} e **k** =4, il metodo stamperà 6 15, poi 6 90 e infine 15 90 (stampa solo 3 coppie perché la quarta non esiste).. Nel caso in cui l'array **a** sia null non va stampato niente.

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[] eliminaCoppie(int[] a)
```

che preso in input un array **a** di elementi interi, restituisce un nuovo array ottenuto da **a** eliminando le coppie di elementi consecutivi uguali.

Ad esempio: se **a**={5, 7, 7, -9, 5, 5, 5, -3} allora viene restituito il nuovo array {5, -9, -3} (vengono eliminati i due valori uguali in posizione 1 e 2 dell'array **a** (entrambi uguali al valore 7) e quelli in posizione 4, 5 e 6 (tutti e tre uguali al valore 5)).

Altro esempio, se **a**={-7, 1, 3, 1, 1,1} il metodo dovrà restituire il nuovo array {-7, 1, 3}.

Se in **a** non ci sono valori consecutivi ripetuti, 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:

```
{37, 26, 32, 44,16, 29, 33, 17, 21, 42, 15}
```

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 valoriMaggiori(int[] a, ink)
```

che restituisce una stringa composta dai valori presenti nell'array **a** che sono strettamente maggiori di **k**. Ad esempio, dato l'array **a**={1,13,-2,8,10,5} e **k** =5 restituisce la stringa "valori maggiori: 13, 8, 10". Se non ci sono valori strettamente maggiori di **k** il metodo restituisce la stringa "non ci sono valori maggiori".

Si sviluppi un metodo Java:

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

che restituisce un nuovo array ottenuto incrementando il primo elemento del valore 0, il secondo del valore 1 e così via. Ad esempio, dato l'array {1,3,5,-2,0,9} restituisce l'array {1,4,7,1,4,14}.