

Appello di Programmazione e Algoritmi 1 26 gennaio 2022 A Distanza

Esercizio 1

Cosa stampa il seguente programma Java?

```
public static void main(String[] args) {
    System.out.println(rebus(-13));
    System.out.println(rebus(17));
    System.out.println(rebus(24));
}
static int rebus (int x){
    if (x<0 || x>=30) return x;
    if (x<=12) return 12+ rebus(-x);
    return x-rebus(x-6);
}
```

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

Cosa stampa il seguente frammento di codice Java?

```
public static void main(String[] args) {
    int valore=0;
    int[] a = {8, 4, 8, 7, 3, 3};
    int n=15, i;
    for (i=a.length-1; i>0 && valore <=2; i--) {
        for (int j=0; j<i; j++) {
            if (a[i]*a[j]<=n){
                valore++;
                System.out.println("i e j " + i + " e " +j);
            }
        }
        System.out.println("valore: " + valore);
    }
}
```

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

Esercizio 2

Scrivere un metodo in Java

```
static int[] eliminaMaggiori (int[] a, int x)
```

che preso in input l'array **a** di elementi interi e l'intero **x**, restituisce un nuovo array in cui sono presenti tutti e soli gli elementi di **a** che sono strettamente minori di **x**, mantenendo l'ordine in cui gli elementi compaiono in **a**. Ad esempio: se **a**={5, 7, -9, 5, 8, 5,10, -3, 8} e **x**=6, il metodo dovrà restituire ={5, -9, 5, 5, -3} (i valori di **a** che sono strettamente minori di 6). Altro esempio, se **a**={1, -2, 3, 1, 5, 2} e **x** =-2, il metodo dovrà restituire {}.

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[] dividiPosNeg (int[] a)
```

che preso in input un array **a** di elementi interi, restituisce un nuovo array della stessa lunghezza di **a** ottenuto mettendo prima gli elementi positivi di **a** e poi gli elementi negativi, nell'ordine in cui compaiono in **a**.

Ad esempio: se **a**= {20, -11, 4, 7, -17, 6} il metodo dovrà restituire {20, 4, 7, 6, -11, -17}.

Altro esempio, se **a**={1, 2, -3, 1, -5, 2} il metodo restituirà {1, 2, 1, 2, -3, -5}.

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:

```
{12, 6, 10, 15,7, 19, 17, 10, 3, 21, 13}
```

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

Si descriva la complessità computazionale del merge-sort (anche scrivendo e risolvendo la ricorrenza $T(n)$).

[Esercizio di Laboratorio] In relazione al progetto di laboratorio

Si sviluppi un metodo Java:

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

che restituisce un nuovo array la cui lunghezza è uguale a quella dell'array **a** diminuita di 1, ottenuto rimuovendo l'elemento in posizione 0. Se l'array **a** è vuoto, restituisce l'array **a**. Ad esempio, dato l'array {1,3,5,-2,0} restituisce l'array {3,5,-2,0}.

Si sviluppi un metodo Java:

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

che restituisce una stringa composta dai valori dell'array **a** ripetuti due volte. Ad esempio, dato l'array {1,3,-2,0} restituisce la stringa "{1,1,3,3,-2,-2,0,0}".