

# PROGRAMMAZIONE II - a.a. 2020-21

## Esercizi – 29 Settembre 2020

### Domande

Si consideri la dichiarazione di classe descritta di seguito

```
class Color {  
    String name;  
    int r = 0, g = 0, b = 0;  
}
```

Si consideri il seguente frammento di codice

```
String s = "Red";  
Color c = new Color();  
c.name = s;  
c.g = 255;  
Object[] a = new Object[2]; 6 a[0] = s;  
a[1] = c;  
/* punto A*/  
Object[] b = a;  
b[1] = new Color[2]; 9 b = (Color[])a[1];  
b[0] = c;  
b[1] = b[0];  
/* punto B */
```

1. Descrivere la struttura della memoria dinamica (heap) nel punto A dell'esecuzione del codice precedente.
2. Descrivere la struttura della memoria dinamica (heap) nel punto B dell'esecuzione del codice precedente.

### Esercizio di programmazione

Si consideri l'interface `IntMultiSet`.

```

public interface IntMultiSet {

    /**
     * Restituisce il numero delle occorrenze dell'intero elem nel multi-insieme.
     * A elem non appartenente al multiinsieme restituisce 0
     */
    int getCount(int elem);

    /**
     * Aggiunge al multi-insieme una occorrenza di elem
     */
    void add(int elem);

    /**
     * Aggiunge n copie di elem al multi-insieme
     */
    void add(int elem, int n);

    /**
     * Rimuove tutte le occorrenze di elem dal multi-insieme.
     * Restituisce false se elem non e' presente
     */
    boolean remove(int elem);

    /**
     * Rimuove n copie di elem dal multi-insieme
     * Restituisce false se elem non e' presente
     */
    boolean remove(int elem, int n);

    /**
     * Restituisce il numero degli elementi del multi-insieme.
     */
    int size();
}

```

1. Si fornisca una implementazione della interface `IntMultiSet` che utilizzi `array` di interi come struttura di realizzazione.
2. Estendere la nozione di e l'implementazione di `IntMultiSet` con operazioni di unione e intersezione.