# Task 1:

# Write a function that accepts two arguments: an array and a number *n*. Assume that the array contains integers. The method should display all of the numbers in the array that are greater than the number *n*.

# You may use the following header for this method:

**void largerThanNumber(int[] array, int number)**

**Note: Declare your array in main() and call respective functions properly and display appropriate messages. You may need to define an input function to fill data in array.**

# Task 2:

# Write a method that returns the index of the nth occurrence of an integer value in an integer array. That is, the method searches for the given value for its nth occurrence in an array, where n is given by user. If that value does not occur n times, the method returns an index of -1.

# You may use the following header for this method:

**int getNthIndex(int[] array, int value, int n)**

For example, the function call:

**int index = getNthIndex(array, 4, 100);**

Searches for the **4th occurrence** in the array for the value **100.** If found, the index of the found value is returned, otherwise **-1** is returned.

**Note: Declare your array in main() and call respective functions properly and display appropriate messages. You may need to define an input function to fill data in array.**

# Task 3:

Write a method which returns index of ***smallest*** and ***largest*** elements in array. As there are more than one returning values, recall the concept of **Pass by Value/ Pass by Reference**. You need to define and implement function by your own.

**For Example:**

Consider an array given below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 10 | 12 | 6 | 10 | 58 | 15 | 1 | 0 | -5 |

**Output:**

Largest Element Index = 5

Smallest Element Index = 9

**Note: Do not write any output anything on console from your defined function. You need to output the answer on console from main(),You may need to define an input function to fill data in array.**

# Task 4:

Assume that the array is of **int** type. Write the following functions:

1. A function called **printElementsAtEvenIndex** that accepts an array of type **int** and prints every element at even index.
2. A function called **printEvenElements** that accepts an array of type **int** and prints every element which is even.
3. A function called **reversePrint** that accepts an array of type **int** and prints the array in reverse order.

**Note: Declare your array in main() and call respective functions properly and display appropriate messages. You may need to define an input function to fill data in array.**