**Every Question should be implemented as a function with input array and size [if given] as parameter, no main and any other work is needed.**

**Question # 1**You have an array **Arr** of N positive integer values; array is divided into three equal parts and every part is sorted either in ascending or descending order [as a whole array may have duplicate values but every part is filled with the unique values]. Your task is to find the minimum value from the array in minimum possible steps. You can consider that array size is in exact multiple of 3. Your solution should be in constant steps, to search min from array is not needed. Think on the paper and then write the code. The Time Complexity of the function shall be **O(1)**.

Example:

Given 🡪2 7 9 12 21 26 33 35 35 24 7 1 [array size is 12, every part is of 4 size]  
Minimum value is 🡪 1

**Question # 2**  
Suppose ***Arr*** is a very large array of integer elements. The ***size*** of ***Arr*** is unknown. The first ***k*** elements are positive integers, greater than 0 and less than 1000, in increasing sorted order. The rest of the elements are greater than 1000.

Write a function that determines whether a given positive integer, ***key***, of value less than 1000 is an element in ***Arr*** or not. If the ***key*** is present in ***Arr***, it returns its ***index*** otherwise it returns **-1**. The Time Complexity of the function shall be **O(log(n))**.

(**Note:** It is to be remembered that both the ***size of Arr*** and ***k*** are ***unknown***.)  
**Array index is starting from 1 not 0 [pseudo code needed]**

Example:

**Arr**: 4 5 7 12 15 24 67 221 233 430 500 1100 1200 1150 1050 1200 …

**key**: 15

returned value: 5

**Arr**: 5 9 9 12 19 29 67 67 120 220 330 400 700 1001 1100 1150 1050 11250 1200 …

**key**: 900

returned value: -1

**Question # 3**

You are given an array **Arr** of n distinct integers sorted in increasing order. Write an **O(lg(n))** function, in PSEUDO-CODE, that finds an index i such **that Arr[i] = i**

where 1<= i <= n. If no such index exists, algorithm returns -1.

**Array index is starting from 1 not 0 [pseudo code needed]**

**Example 1:**

Arr: 2, 3, 5, 7, 9, 11, 23, 34, 45, 67

function returns: -1

**Example 2:**

Arr: -9, -7, -5, -3, -2, 1, 3, 4, 6, 10

function returns: 10