

# Array algorithms

Chương trình sau bao gồm một số giải thuật cơ bản về Array, em hãy hoàn thiện các giải thuật này và chương trình kiểm tra.

## Input

Dòng đầu tiên gồm hai số nguyên tương ứng là chiều dài của dãy số và số câu lệnh. Dòng thứ hai là dãy số

```
5 5
5 2 7 3 5
findMin
findMax
average
indexOf 3
lastIndexOf 5
```

## Output

```
2
7
4.4
3
4
```

## Program

```
import java.util.Scanner;

public class ArrayAlgorithms {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();
        int m = sc.nextInt();
        int[] numbers = new int[n];
        for (int j = 0; j < n; j++) {
            numbers[j] = sc.nextInt();
        }

        for (int i = 0; i < m; i++) {
            String command = sc.next();

            long result = 0;
            switch (command) {
                case "findMax":
                    result = findMax(numbers);
                    break;
                case "findMin":
                    result = findMin(numbers);
                    break;
            }
            System.out.println(result);
        }
    }
}
```

// Add your code here

```

// Add your code here

}
}

static int findMax(int[] numbers) {
// Add your code here
return 0;
}

static int findMin(int[] numbers) {
// Add your code here
return 0;
}

static double average(int[] numbers) {
// Add your code here
return 0;
}

static int sum(int[] numbers) {
// Add your code here
return 0;
}

/**
 * @Description:
 * @Hint: find maximum number in the array, then check which number is
 */
static int countMax(int[] numbers) {
// Add your code here
return 0;
}

/**
 * @Description: sum all odd numbers in array
 * @Hint: what is odd number, how to check?
 */
static int sumOddNumber(int[] numbers) {
// Add your code here
return 0;
}

static int countEvenNumber(int[] numbers) {
// Add your code here
return 0;
}

static int indexOf(int[] numbers, int value) {
// Add your code here
return -1;
}

static int lastIndexOf(int[] numbers, int value) {
// Add your code here
return -1;
}

/**
 * @Description: find minimum difference between two consecutive numbers in an
 * array
 * @Hint: minimum of Math.absolute(array[i+1] - array[i])
 */
static int findMinDistance(int[] numbers) {
// Add your code here

```

```
// Add your code here
return 0;
}

/**
 * @Description: Check if first number equal last number, second number equal
 *               second to last number, and so on...
 * @Hint: i-th number equal (length-1-i)-th number
 */
static boolean isPalindrome(int[] numbers) {
// Add your code here
return false;
}

/**
 * @Description: Check if number is in array
 */
static boolean contains(int[] numbers, int number) {
// Add your code here
return false;
}

/**
 * @Description: Check if array has duplicated numbers
 * @Hint: Should check every pair of number, array[i] vs. array[j], i != j
 */
static boolean hasDuplicated(int[] numbers) {
// Add your code here
return false;
}

/**
 * @Description: Check if array has two numbers A and B where A % B = 0
 * @Hint: Should check every pair of number, array[i] vs. array[j], i != j
 */
static boolean hasDivisor(int[] numbers) {
// Add your code here
return false;
}
}
```