MOHANA M 2023-CSCS-A M2 • REC-CIS

# **CS23333-Object Oriented Programming Using Java-2023**

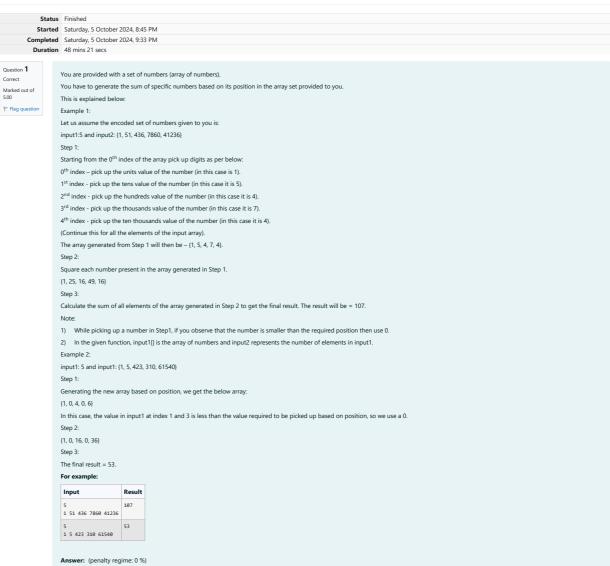
Question 1

Marked out of 5.00

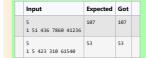
Correct

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-03-Arrays / Lab-03-Logic Building





```
1 - import java.util.Scanner;
                ublic class NumberArrayProcessor {
                    public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
                            int size = scanner.nextInt();
    // Get the array elements from the user
int[] array = new int[size];
                            for (int i = 0; i < size; i++) {
    array[i] = scanner.nextInt();</pre>
                            // Process the array to get the result
int result = processArray(size, array);
System.out.println( result);
                    public static int processArray(int size, int[] array) {
   int[] extractedValues = new int[size];
                              // Step 1: Extract specific digits based on the position
for (int i = 0; i < size; i++) {
    extractedValues[i] = extractDigit(array[i], i);
}</pre>
                             // Step 2: Square each extracted value
for (int i = 0; i < size; i++) {
    extractedValues[i] = extractedValues[i] * extractedValues[i];</pre>
                             // Step 3: Calculate the sum of the squared values
int sum = 0;
for (int value : extractedValues) {
    sum += value;
                     public static int extractDigit(int number, int position) {
    // Extract the digit at the given position from the ri,
    for (int i = 0; i < position; i++) {
        number /= 10;
    }
}</pre>
                                                                                                                                                                                                                                                                                                                                      ┙
```



Passed all tests!

Question **2**Correct
Marked out of 5.00

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array

If there are NO positive numbers in the array, you are expected to return -1.

In this question's scope, the number 0 should be considered as positive.

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below).

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13, 32, 34, -5, 66, 78, 78, -79}

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = 12 + 18 + 18 + 14 = 63.

Example 2:

input1 = 11

input2 = {-22, -24, 16, -1, -17, -19, -37, -25, -19, -93, -61}

Expected output = -1

Explanation

There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

Example 3:

input1 = 16

input2 = {-58, 32, 26, 92, -10, -4, 12, 0, 12, -2, 4, 32, -9, -7, 78, -79}

Expected output = 174

xplanation:

The input array contains four sequences of POSITIVE numbers, i.e. "32, 26, 92", "12, 0, 12", "4, 32", and "78". The first and second sequences "32, 26, 92" and "12, 0, 12" are the longest of the four as they contain 4 elements each. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = (32 + 26 + 92) + (12 + 0 + 12) = 174.

#### For example:

Input	Result
16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62
11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1
16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174

```
Answer: (penalty regime: 0 %)
```

```
| Input | Expected | Got | 16 | -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79 | 62 | 62 | | 11 | -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61 | 174 | 174 | -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79 | 174 | 174 |
```

## Passed all tests!

Question 3
Correct
Marked out of 5.00
Flag question

Given an integer array as input, perform the following operations on the array, in the below specified sequence

- Find the maximum number in the array
- 2. Subtract the maximum number from each element of the array.
- Multiply the maximum number (found in step 1) to each element of the resultant array.

After the operations are done, return the resultant array.

Example 1:

input1 = 4 (represents the number of elements in the input1 array)

```
input2 = {1, 5, 6, 9}
Expected Output = {-72, -36, 27, 0}
Explanation:
Step 1: The maximum number in the given array is 9.
Step 2: Subtracting the maximum number 9 from each element of the array.
\{(1-9), (5-9), (6-9), (9-9)\} = \{-8, -4, -3, 0\}
Step 3: Multiplying the maximum number 9 to each of the resultant array:
\{(-8 \times 9), (-4 \times 9), (3 \times 9), (0 \times 9)\} = \{-72, -36, -27, 0\}
So, the expected output is the resultant array {-72, -36, -27, 0}.
Example 2:
input1 = 5 (represents the number of elements in the input1 array)
input2 = {10, 87, 63, 42, 2}
Expected Output = {-6699, 0, -2088, -3915, -7395}
Explanation
Step 1: The maximum number in the given array is 87.
Step 2: Subtracting the maximum number 87 from each element of the array:
{(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)} = {-77, 0, -24, -45, -85}
Step 3: Multiplying the maximum number 87 to each of the resultant array:
\{(-77\times87),\,(0\times87),\,(-24\times87),\,(-45\times87),\,(-85\times87)\} = \{-6699,\,0,\,-2088,\,-3915,\,-7395\}
So, the expected output is the resultant array {-6699, 0, -2088, -3915, -7395}.
Example 3:
input1 = 2 (represents the number of elements in the input1 array)
input2 = {-9, 9}
Expected Output = {-162, 0}
```

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

{(-9 - 9), (9 - 9)} = {-18, 0}

Step 3: Multiplying the maximum number 9 to each of the resultant array:

 $\{(-18 \times 9), (0 \times 9)\} = \{-162, 0\}$ 

So, the expected output is the resultant array {-162, 0}.

Note: The input array will contain not more than 100 elements

### For example:

Input	Result
4 1 5 6 9	-72 -36 -27 0
5 10 87 63 42 2	-6699 0 -2088 -3915 -7395
2 -9 9	-162 0

### Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
           int n = scanner.nextInt();
int[] arr = new int[n];
  for (int i = 0; i < n; i++) {
    arr[i] = scanner.nextInt();</pre>
                      int maxElement = findMaxElement(arr);
                      // Calculate arr1 and arr2
int[] arr2 = new int[n];
for (int i = 0; i < n; i++) {
    int arr1 = arr[i] - maxElement;
    arr2[i] = arr1 * maxElement;
                      for (int i = 0; i < n; i++) {
    System.out.print(arr2[i] + " ");</pre>
                       }
System.out.println();
                      // Close the scar
scanner.close();
                private static int findMaxElement(int[] arr) {
                      int maxelement = arr[0];
for (int i = 1; i < arr.length; i++) {
    if (arr[i] > maxelement) {
        maxelement = arr[i];
    }
                       return maxElement;
```

```
Input
                  Expected
                                             Got
                  -72 -36 -27 0
                                              -72 -36 -27 0
   5
10 87 63 42 2
                   -6699 0 -2088 -3915 -7395 -6699 0 -2088 -3915 -7395
                   -162 0
                                              -162 0
Passed all tests!
```

Finish review