# 02 Fundamentals

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++03 or the C++11 standard.

Please follow the exact instructions on uploading the solutions for each task.

Check your solutions here: <https://judge.softuni.bg/Contests/Practice/Index/1349#0>

# Task 1 – Compare Arrays

Write a program that reads two Integers array from the console and compares them element by element. For better code reusability, you could do the comparison in a

bool areEqual(int arr1[], int length1, int arr2[], int lenght2)

function, which returns TRUE if they are equal and FALSE if not.

Each array will be defined by two lines on the console – the first containing a number representing the length of the array, and the second containing the numbers in the array.

Print **equal** if the arrays match, and **not equal** if the arrays don’t match

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1 2 3  2  2 1 | Not equal |

# Task 2 – Longest Sequence

Write a program that find the longest sequence of equal elements in an integer array and then prints that sequence on the console ( integers separated by space on a single line ).

if there is more than one such sequence, print the last one. The input array will be entered on two lines – the first line will contain an integer representing the number of elements, the second will contain the elements separated by spaces.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7  13 10 10 1 4 2 10 | 4 |

# Task 3 – Above Average

Write a program that read an array of integer numbers from the console and prints all numbers which are larger than or equal to the mathematical average of the numbers in the array. The output should be printed on a single line, separating the output number by spaces. The output numbers should be in the same order as they were in the input.

The output array will be entered on two lines- the first line will contain an integer representing the number of elements, the second will contain the elements separated by spaces,.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  1 2 3 4 5 | 3 4 5 |

# Task 4 – Most Frequent Number

Write a program that find the most frequent number in a given sequence of numbers.

Numbers will be in the range [0, 9].

In case of multiple numbers with the same maximal frequent, print all of them, ordered, from smallest to largest, separated by spaces.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Output** |
| 13  4 1 1 4 2 3 4 4 1 2 4 9 3 | 4 | The number 4 is the most frequent |
| 8  2 2 2 2 1 2 2 2 | 2 | The number 2 is the most frequent |

# Task 5 – Cartesian Product

Write a program which reads an array from the console and prints the product of each of its elements with all elements. E.g. for the array {1, 7, 3}, the result would be

{1\*1, 1\*7, 1\*3, 7\*1, 7\*7, 7\*3, 3\*1, 3\*7, 3\*3},

which gives us the array

{1, 7, 3, 7, 49, 21, 3, 21, 9},

so for the input

1 7 3

the program should print

1 7 3 7 49 21 3 21 9

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1 7 3 | 1 7 3 7 49 21 3 21 9 |
| 2  -1 4 | 1 -4 -4 16 |

# Task 6 – Closest Numbers

Write a program which finds the two closest (by value) integer numbers in an array and prints the absolute difference between them.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Output** |
| 5  1 105 10 100 3 | 2 | The closest numbers are 1 and 3, abs(1,3) = 2 |
| 9  1 2 3 4 5 6 7 8 9 | 1 | All numbers are exactly 1 unit apart |