

SoftUni Judge System Guidelines

This document describes how assignments are sent and verified through the fully automated “SoftUni Judge” system. The system is online at <https://judge.softuni.bg>. The username and the password are the ones used for logging into <https://softuni.bg>.

Fully automated testing system

Submitting exercises is fully automated and online. It is done through “SoftUni Judge”: <https://judge.softuni.bg>.

- The assignments are checked online with the “SoftUni Judge” tool through a series of tests. Each successfully passed test brings points to the overall score for the assignment. A test is passed successfully when its result is correct and the testing time is within certain limits.
- **The tests** used by “SoftUni Judge” for verifying assignments are not revealed during competition mode.
- Each participant uses his/her **username** and **password** for <https://softuni.bg> to enter the judge tool.
- Submitting assignments and their verification happens **in real time**. Once an assignment is submitted, the judge tool responds in seconds with the following verification result:
 - The amount of **points** the participant gets from the submitted solution – between 0 and 100
 - A compile time **error message**
- The participant receives the following **status information** for every **test**:
 - Correct result
 - Wrong result
 - Runtime error
 - Time limit
 - Memory limit
- The judge tool verifies the output from the tests **symbol by symbol**
 - Each comma, unnecessary symbol or a missing whitespace results in **0 points** for the corresponding test.
 - Please **do not include any unwanted information** to your assignments, such as “Please enter N=” when it is required to enter a number as an input. This will bring **0 points**.
 - If the output requires a number to be printed to the console (for example: 25), do not include any descriptive messages, such as “The result is 25”. Print **only what is asked in the assignment**.
- The system supports **public rankings** in real time, accessible to all SoftUni students.
 - The rankings display the points per assignment per student.
- The highest score achieved for every assignment is kept in the rankings. If a participant submits a solution that scores less than the solutions he/she has sent before, the system will not take points away.

Programming Languages

The judge system supports the following programming languages:

- **C# 7** – Microsoft Visual C# Compiler version 1.0.0.50618, 64-bit on Windows, C# version 7
- **Java 8** – **javac** 1.8.0 Compiler, 64-bit on Windows
- **C 99** – GCC 5.2 on Windows, MinGW-w64 environment
- **C++ 14** – GCC 5.2 on Windows, MinGW-w64 environment
- **JavaScript** – Node.JS version v8.10.0, 64-bit on Windows (V8 engine, ECMA-262, 3rd edition)
- **Python 3.6** – Python version 3.6.0, 64-bit on Windows
- **PHP 7** – PHP version 7.1.11 (CLI), 64-bit on Windows

An Example Assignment – Min3Numbers

You can test your solution for the Min3Numbers exercise with the judge tool at:

<https://judge.softuni.bg/Contests/Practice/Index/132>.

You are given an **N amount of numbers**: a_0, a_1, \dots, a_{N-1} . Find the three numbers with the smallest values and print them on the console.

Input

- The input consists of the following lines:
 - First line: You will receive **N** – the amount of numbers
 - N number of lines: **Each line has one number**

On the first line of the input you will receive **N** – the amount of numbers. On the next N number of lines there will be **one number per line**. The input data will be correct and within the described format. There is no need to verify the input.

Output

Print out to the console **the three smallest numbers** in increasing order. Print each number on a new line. If the numbers are less than three, print them anyway in increasing order.

Constraints

- N** is an integer within the range of [1 ... 10 000].
- The numbers **a0, a1, ..., aN-1** are integers in the range of [-100 000 ... 100 000].
- The **time limit** is 100 ms
- The **memory limit** is 16 MB

Examples

You can find examples of inputs and their corresponding outputs in the tables below:

Input	Output
5	-5
50	10
10	15
30	
15	
-5	

Input	Output
2	111
222	222
111	

Input	Output
1	20
20	

Input	Output
6	-6
-1	-5
-2	-4
-3	
-4	
-5	
-6	

Input	Output
3	1
1	2
2	3
3	

Scroll down to see implementations in different languages.

C# Solution

This is an example solution with C#. The standard C# console is used for the input and the output.

Min3Numbers.cs

```
using System;
using System.Linq;

class Min3Numbers
{
    static void Main()
    {
        int n = int.Parse(Console.ReadLine());
        int[] numbers = new int[n];
        for (int i = 0; i < n; i++)
        {
            numbers[i] = int.Parse(Console.ReadLine());
        }

        var smallest3Nums = numbers.OrderBy(i => i).Take(3);

        foreach (var num in smallest3Nums)
        {
            Console.WriteLine(num);
        }
    }
}
```

Constraints in the judge system about the C# language:

- Supported version: C# 7, Microsoft Visual C# Compiler, 64-bit on Windows.
- In case multiple classes are implemented, they must be all placed inside of one file, one after another. There may be only one **Main()** method.
- Libraries outside the .NET Framework 4.7 standard are NOT to be used.
- Only the [Wintellect.PowerCollections](#) library is accepted.

C++ Solution

This is an example solution with C++. The standard C++ console is used for the input and the output.

Min3Numbers.cpp

```
#include <vector>
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> numbers;
    for (int i = 0; i < n; i++) {
        int num;
        cin >> num;
        numbers.push_back(num);
    }

    sort(numbers.begin(), numbers.end());

    int count = 0;
    for (auto it = numbers.begin(); it != numbers.end(); ++it) {
        cout << *it << endl;
        count++;
        if (count >= 3) {
            return 0;
        }
    }

    return 0;
}
```

Constraints in the judge system about the C++ language:

- Supported version: C++ 14, GCC 5.2 on Windows (MinGW-w64).
- Libraries outside the standard C++ STL are NOT to be used.
- The type **long** is 32 bits.

C Solution

This is an example solution with C. The standard C console is used for the input and the output.

Min3Numbers.c

```
#include <stdio.h>
#include <stdlib.h>

int int_compare(const void *a, const void *b) {
    return (*(int*)a - *(int*)b);
}

int main() {
    int n;
    scanf("%d", &n);
    int* numbers = (int*)malloc(sizeof(int) * n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &numbers[i]);
    }

    qsort(numbers, n, sizeof(int), int_compare);

    int count = (n < 3) ? n : 3;
    for (int i = 0; i < count; i++) {
        printf("%d\n", numbers[i]);
    }

    return 0;
}
```

Constraints in the judge system about the C language:

- Supported version: C99, GCC 5.2 on Windows (MinGW-w64).
- Libraries outside the standard library are NOT to be used.
- The type **long** is 32 bits.

Java Solution

This is an example solution with Java. The standard Java console is used for the input and the output.

Min3Numbers.java

```
import java.util.Arrays;
import java.util.Scanner;

public class Min3Numbers {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        int[] numbers = new int[n];
        for (int i = 0; i < n; i++) {
            numbers[i] = scan.nextInt();
        }

        Arrays.sort(numbers);

        for (int i = 0; i < Math.min(n, 3); i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

Constraints in the judge system about the Java language:

- Supported version: Java 8, javac 1.8.0 Compiler, 64-bit on Windows
- Libraries outside the standard JDK 8 library are NOT to be used.
- There must be only one public class with a **main(args)** method.
- In case of the implementation of various classes, they must be all placed inside of the source code, one after another. It is required that only one of them is public.

JavaScript Solution

This is an example solution with JavaScript. The input from the function `solve(arr)` is an array of strings. The output is to be printed on the console.

Min3Numbers.js

```
function solve(arr) {
  var numbers = arr.splice(1).map(Number);

  numbers.sort(function (a, b) { return a - b; });
  let smallest3Numbers = numbers.slice(0, 3);

  for (let num of smallest3Numbers) {
    console.log(num);
  }
}
```

Constraints in the judge system about the JavaScript language:

- Supported version: JavaScript on Node.JS version 8.10.0, 64-bit on Windows (V8 engine, ECMA-262, 3rd edition)
- Submitted only one function `solve(arr)`, which receives the input as an array of strings.
- If multiple functions are used, they must be all placed inside the main function.
- Printing on the console is to be done with `console.log(...)`.

Python Solution

This is an example solution with Python. The standard Python console is used for the input and the output.

Min3Numbers.py

```
n = int(input())
nums = list()
for i in range(0, n) :
    nums.append(int(input()))
nums = sorted(nums)
count = min(len(nums), 3)
for i in range(0, count) :
    print(nums[i])
```

Constraints in the judge system about the Python language:

- Supported version: Python 3.6.0, 64-bit on Windows
- The input is to be read with the standard `input()` or with `sys.stdin`.
- The output is to be printed with the standard `print()` or with `sys.stdout`.

PHP Solution

This is an example solution with PHP. The standard CLI console is used for the input and the output.

Min3Numbers.php

```
<?php
    fscanf(STDIN, "%d", $n);
    $numbers = array($n);
    for ($i=0; $i < $n; $i++) {
        fscanf(STDIN, "%d", $numbers[$i]);
    }

    sort($numbers);
    $smallest3Numbers = array_slice($numbers, 0, 3);

    foreach ($smallest3Numbers as $num) {
        fprintf(STDOUT, "%d\n", $num);
    }
?>
```

Constraints in the judge system about the PHP language:

- It is supported: PHP 7 CLI (command line interface), engine PHP version 7.1.11, 64-bit on Windows
- The input is read from the standard input – a file with a name **STDIN**.
- The result is printed with the standard output – a file with a name **STDOUT**.

Link to the SoftUni Judge system – Min3Numbers

You can test your solution for the Min3Numbers exercise with the judge tool at:

<https://judge.softuni.bg/Contests/Practice/Index/132>.

Submitting a solution

Once you have logged-in at SoftUni Judge, submitting a solution is done from your user interface:

The screenshot shows the submission interface for the 'Min 3 Numbers' problem. The browser address bar shows the URL <https://judge.softuni.bg/Contests/Practice/Index/132#0>. The page title is 'Submit a solution'. The problem name 'Min 3 Numbers' is highlighted with a red box. Below the problem name, there are links for 'Condition of the task', 'Solution of C#', 'C++ Solution', 'Java solution', 'JavaScript Solution', 'Decision of C', 'Solution in PHP', 'Python Solution', and 'Ruby's decision'. A code editor is visible with a black background. Below the code editor, there are submission limits: 'Allowed working time: 0.100 sec.', 'Allowed memory: 16.00 MB', 'Size limit: 16.00 KB', and 'Checker: Trim'. A dropdown menu is open, showing options for 'C# code', 'C++ code', 'JavaScript code (NodeJS)', 'Java code', 'PHP code (CLI)', and 'Python code'. The 'Submit' button is highlighted with a red box. To the right, there is a 'Problem results' table with columns 'Participant' and 'Result'. The table contains the following data:

Participant	Result
Hofhearted	100 / 100
boryana.ai	100 / 100
moni200001	100 / 100
ObedMarsh	100 / 100

Results

The results from the submitted solutions appear in the table below the submit form a couple of seconds after sending them:

The screenshot shows the 'Submissions' table for the 'Min 3 Numbers' problem. The browser address bar shows the URL <https://judge.softuni.bg/Contests/Practice/Index/132#0>. The table has columns 'Points', 'Time and memory used', and 'Submission date'. The table contains the following data:

Points	Time and memory used	Submission date
100 / 100	Memory: 10.04 MB Time: 0.022 s	15:39:11 27.10.2015
70 / 100	Memory: 9.93 MB Time: 0.151 s	15:37:41 27.10.2015
40 / 100	Memory: 13.47 MB Time: 0.054 s	14:15:25 27.10.2015
20 / 100	Memory: 1.59 MB Time: 0.002 s	12:13:21 27.10.2015
20 / 100	Memory: 1.96 MB Time: 0.024 s	11:50:59 27.10.2015
Compile time error	---	11:49:42 27.10.2015