

# Ръководство за използване на SoftUni Judge системата

В настоящото ръководство даваме насоки как се използва автоматизираната система „SoftUni Judge“ за изпращане и проверка на задачи по програмиране. Системата е достъпна от <https://judge.softuni.bg> с потребител и парола, същите като за <https://softuni.bg>.

## Система за автоматизирано тестване

Предаването на задачи за автоматизирано тестване се извършва **през Интернет** от сайта на **judge системата**: <https://judge.softuni.bg>.

- Задачите се тестват автоматизирано чрез **online judge система** чрез поредица от **тестове**. Всеки успешно преминат тест дава предвидените за него **точки** (когато е получен верен резултат при в рамките на позволеното време).
- **Тестовите**, с които judge системата проверява задачите, не се разкриват по време на състезание.
- Всеки участник влиза в judge системата със своя **потребител и парола** за сайта на СофтУни.
- Предаването на задачи и проверката стават в **реално време**. След изпращане на решение системата връща резултат след няколко секунди:
  - Брой присъдени **точки** (между 0 и 100), когато предаденият код се компилира успешно.
  - Или **съобщение за грешка** по време на компилация.
- За всеки от тестовите се получава информация **как е преминал теста**:
  - Правилен резултат (correct answer).
  - Грешен резултат (wrong answer).
  - Грешка / изключение по време на изпълнение (runtime error).
  - Превишено време (time limit).
  - Превишена памет (memory limit).
- Изходните резултати се проверяват от системата **символ по символ**.
  - Всяка запетайка, **излишен символ** или липсващ интервал носи **0 точки** на съответния тест.
  - Моля **не извеждайте излишна информация**, например текстове като „Please enter N =“, когато се изисква въвеждане на единично число. Това води до **0 точки**.
  - Ако в изхода се изисква да се отпечата число (примерно 25), **не извеждайте описателни съобщения** като „The result is 25“, а отпечатайте точно каквото се изисква.
- Системата поддържа **публично временно класиране** в реално време, достъпно за всички участници.
  - В класирането се вижда всеки участник какви точки има на всяка от задачите.
- За всяка задача системата пази **най-високият постигнат резултат**. Ако пратите по-лошо решение от предишното ви изпратено, системата няма да ви отнеме точки.

## Езици за програмиране

Автоматизираната judge система поддържа следните езици за програмиране:

- **C# 7** – CSC версия 2.10.0.0, x64 Windows, C# версия 7.3
- **Java 11** – OpenJDK 11.0.13, x64 Ubuntu
- **C 99** – компилатор GCC 7.5.0, x64 Ubuntu
- **C++ 11** – компилатор GCC 7.5.0, x64 Ubuntu
- **JavaScript** – Node.JS версия v12.22.9, x64 Ubuntu
- **Python 3.6** – Python версия 3.6.9, x64 Ubuntu
- **PHP 7** – PHP версия 7.2.24 (CLI), x64 Ubuntu

## Примерна задача за тренировка – Min3Numbers

Дадени са **N** числа:  $a_0, a_1, \dots, a_{N-1}$ . Да се намерят и отпечатаат **трите най-малки** от тях.

### Входни данни

На първия ред от входа стои числото **N** – брой числа. На следващите **N** реда стои по **едно число**. Входните данни са коректни и в описания формат и не е необходимо да ги проверявате.

### Изходни данни

Изведете намерените **три най-малки числа**, в нарастващ ред, по едно на ред. Ако числата са по-малко от три, изведете колкото числа има, подредени от най-малкото към най-голямото.

### Ограничения

- **N** е цяло число в диапазона  $[1 \dots 10\,000]$ .
- Числата  $a_0, a_1, \dots, a_{N-1}$  са цели, в диапазона  $[-100\,000 \dots 100\,000]$ .
- **Време** за работа (time limit): 100 ms
- Лимит **памет**: 16 MB

### Примерен вход и изход

В таблиците по-долу са дадени примерни входове за програмата и съответните им изходи:

вход	изход	вход	изход	вход	изход	вход	изход	вход	изход
5	-5	2	111	1	20	6	-6	3	1
50	10	222	222	20		-1	-5	1	2
10	15	111				-2	-4	2	3
30						-3		3	
15						-4			
-5						-5			
						-6			

### Решение на C#

Примерно решение на езика C#. Входът и изходът са от стандартната конзола:

Min3Numbers.cs
<pre>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 &lt; n; i++)         {             numbers[i] = int.Parse(Console.ReadLine());         }     } }</pre>

```

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

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

```

Ограничения за езика C# в judge системата:

- Поддържа се стандарта C# 7, компилатор Microsoft Visual C# Compiler, 64-bit on Windows.
- Ако се използват няколко класа, трябва да се поставят в един файл, един след друг. Може да има само един **Main()** метод.
- Не може да се ползват външни библиотеки извън стандартните за .NET Framework 4.6.
- Може да се използва външната библиотека [Wintellect.PowerCollections](https://www.nuget.org/packages/Wintellect.PowerCollections/).

## Решение на C++

Примерно решение на езика C++. Входът и изходът са от стандартната конзола:

Min3Numbers.cpp
<pre> #include &lt;vector&gt; #include &lt;iostream&gt; #include &lt;algorithm&gt;  using namespace std;  int main() {     int n;     cin &gt;&gt; n;     vector&lt;int&gt; numbers;     for (int i = 0; i &lt; n; i++) {         int num;         cin &gt;&gt; num;         numbers.push_back(num);     }      sort(numbers.begin(), numbers.end());      int count = 0;     for (auto it = numbers.begin(); it != numbers.end(); ++it) {         cout &lt;&lt; *it &lt;&lt; endl;         count++;         if (count &gt;= 3) {             return 0;         }     }      return 0; } </pre>

Ограничения за езика C++ в judge системата:

- Поддържа се стандарта C++ 14, компилатор GCC 6.3.0 on Windows (MinGW-w64).
- Не може да се ползват външни библиотеки извън стандартната за C++ библиотека STL.

- Типът **long** е 32-битов.

## Решение на C

Примерно решение на езика C. Входът и изходът са от стандартната конзола:

### 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;
}
```

Ограничения за езика C в judge системата:

- Поддържа се стандарта C99, компилатор GCC 6.3.0 on Windows (MinGW-w64).
- Не може да се ползват външни библиотеки извън стандартната за C библиотека.
- Типът **long** е 32-битов.

## Решение на Java

Примерно решение на езика Java. Входът и изходът са от стандартната конзола:

### 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]);
}
}
}

```

Ограничения за езика Java в judge системата:

- Поддържа се стандарта **Java 8**, компилатор **javac 1.8.0**, 64-bit on Windows
- Не може да се ползват външни библиотеки извън стандартните в JDK 8.
- Трябва да имате точно един публичен клас с **main(args)** метод.
- Може да се ползват няколко класа, разположени в един след друг в сорс кода, но се изисква само един от тях да е публичен.

## Решение на JavaScript

Примерно решение на езика JavaScript. Входът се подава като масив от стрингове във функцията **solve(arr)**. Изходът се печата на конзолата:

Min3Numbers.js
<pre> function solve(arr) {     var numbers = arr.splice(1).map(Number);      numbers.sort(function (a, b) { return a - b; });     var smallest3Numbers = numbers.slice(0, 3);      for (var i in smallest3Numbers) {         console.log(smallest3Numbers[i]);     } } </pre>

Ограничения за езика JavaScript в judge системата:

- Поддържа се стандарта **JavaScript 1.5** върху платформа Node.JS version v6.9.1, 64-bit on Windows (V8 engine, ECMA-262, 7th edition).
- Трябва да се дефинира точно една функция **solve(arr)**, която приема входните данни като масив от стрингове (по един стрингов елемент за всеки ред от входа).
- Ако се ползват няколко функции, те трябва да са вложени вътре в главната функция.
- Резултатът трябва да се отпечата на конзолата с **console.log(...)**.

## Решение на Python

Примерно решение на езика Python. Входът и изходът са от стандартната конзола:

Min3Numbers.py
<pre> n = int(input()) nums = list() for i in range(0, n) :     nums.append(int(input())) nums = sorted(nums) </pre>

```
count = min(len(nums), 3)
for i in range(0, count) :
    print(nums[i])
```

Ограничения за езика Python в judge системата:

- Поддържа се **Python 3.5.1**, 64-bit on Windows.
- Входните данни с четат от стандартния вход (с **input()** чрез **sys.stdin**).
- Резултатът се печата на стандартния изход (с **print()** или чрез **sys.stdout**).

## Решение на PHP

Примерно решение на езика PHP. Входът и изходът са от стандартната CLI конзола (**STDIN** и **STDOUT**):

### 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);
}
?>
```

Ограничения за езика PHP в judge системата:

- Поддържа се **PHP 7 CLI** (command line interface), engine PHP version 7.0.6, 64-bit on Windows.
- Входните данни с четат от стандартния вход (файл с име **STDIN**).
- Резултатът се печата на стандартния изход (файл с име **STDOUT**).

## Решение на Ruby

Примерно решение на езика Ruby. Входът и изходът са от стандартната конзола (**gets**):

### Min3Numbers.rb

```
n = gets.chomp.to_i
ary = []
for i in 0..n-1
    number = gets.chomp.to_i
    ary.push(number)
end
ary.sort! {|x,y| x <=> y}
ary.take(3).each { |e| print e, "\n" }
```

Ограничения за езика **Ruby** в judge системата:

- Поддържа се **Ruby 2.4.1**, 64-bit on Windows.
- Входните данни с четат от стандартния вход (чрез **gets**).
- Резултатът се печата на стандартния изход (чрез **print**).

## Линк SoftUni Judge системата – Min3Numbers

Задачата за намиране на **трите най-малки числа** е достъпна за тренировка и тестване в SoftUni Judge системата: <https://judge.softuni.bg/Contests/Practice/Index/132>.

## Изпращане на решение

Изпращането на решение става през потребителския интерфейс в SoftUni judge:

The screenshot shows the 'Submit a solution' page 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'. Below the title, there is a navigation bar with 'Min 3 Numbers' and a 'Добави задача' button. The main heading is 'Min 3 Numbers'. Below it, there are buttons for 'Участници', 'Тестове', 'Промяна', and 'Изтриване'. A horizontal menu shows links to 'Условиена задачата', 'Решение на C#', 'Решение на C++', 'Решение на Java', 'Решение на JavaScript', 'Решение на C', 'Решение на PHP', and 'Администрация'. The main area contains a code editor with C# code for the 'Min3Numbers' class. The code is as follows:

```
1 class Min3Numbers
2 {
3     static void Main()
4     {
5         int n = int.Parse(Console.ReadLine());
6         int[] numbers = new int[n];
7         for (int i = 0; i < n; i++)
8         {
9             numbers[i] = int.Parse(Console.ReadLine());
10        }
11        var smallest3Nums = numbers.OrderBy(i => i)
12        foreach (var num in smallest3Nums)
13        {
14            Console.WriteLine(num);
15        }
16    }
17 }
18
19
20
```

Below the code editor, there are constraints: 'Allowed working time: 0.10 sec.', 'Allowed memory: 16.00 MB', 'Size limit: 16.00 KB', and 'Checker: Trim'. A dropdown menu is open, showing options: 'C# code', 'C++ code', 'JavaScript code (NodeJS)', 'Java code', and 'PHP code (CLI)'. The 'C# code' option is selected. A 'Submit' button is visible to the right of the dropdown.

## Проверка на резултат

Резултатите от автоматизирано тестване на изпратените за проверка решения се появяват в таблицата под формата за изпращане на решение, няколко секунди по-късно:

Allowed working time: 0.10 sec.  
Allowed memory: 16.00 MB  
Size limit: 16.00 KB  
Checker: Trim

C# code

Submit

Submissions			
Points	Time and memory used	Submission date	
✓✓✓✓✓✓✓✓✓✓ 100 / 100	Memory: 10.04 MB Time: 0.022 s	15:39:11 27.10.2015	<a href="#">Details</a>
✓✓⌚✓✓⌚✓✓⌚ 70 / 100	Memory: 9.93 MB Time: 0.151 s	15:37:41 27.10.2015	<a href="#">Details</a>
✗✗✗✓✓✗✗✓✓✗ 40 / 100	Memory: 13.47 MB Time: 0.054 s	14:15:25 27.10.2015	<a href="#">Details</a>
✓✗✗✗✓✗✗✗✗✗ 20 / 100	Memory: 1.59 MB Time: 0.002 s	12:13:21 27.10.2015	<a href="#">Details</a>
✗✗✗✓✓✗✗✗✗✗ 20 / 100	Memory: 1.96 MB Time: 0.024 s	11:50:59 27.10.2015	<a href="#">Details</a>
Compile time error	---	11:49:42 27.10.2015	<a href="#">Details</a>