

stepik

Вот тут можете потренироваться: [ССЫЛКА.](#)

Олимпиада по
скоростному кодированию
Прогресс по курсу: 12/12

1 Junior

1.1 Таблицы

1.2 Графика

1.3 Строки

1.4 Арифметика

Формат
Правила
Ограничения

SpeedCoding

- Соревнования **личные** (не командные).
- Автоматизированная система проверки на платформе **Stepik.org** (вам требуется иметь свой аккаунт на Stepik.org).
- Язык программирования выбираете сами (список ниже).
- Всего **15 задач**, продолжительность - **90 минут**.
- Для каждой задачи отводится некоторое количество разрешённых попыток (зависит от сложности задачи). Штрафы на неуспешную попытку подачи решения не накладываются. **Это даёт возможность идти на некоторый риск и подавать не до конца проверенные решения, тем более на данной платформе вы можете писать код непосредственно на странице задачи.**
- Задачи разбиты по модулям (разделам, темам).
- Всего 5 разделов, в каждом разделе по 3 задачи.
- Разделы 1, 2, 3, 4, как правило, содержат достаточно простые задачи - на реализацию - каждую можно выполнить в течение 5-10 минут.
- Раздел №5 содержит задачи, требующие понимания некоторых специализированных алгоритмов и структур данных - каждую можно выполнить за 10-30 минут.
- Перед началом соревнований вы получаете ссылку на страницу соревнования от организатора. По истечении времени соревнования доступ к подаче решений будет закрыт.

Пример оформления задачи:

Простая сумма

Напишите программу, которая считает **сумму двух чисел**.

На **вход** подаётся одна строка, в которой через пробел написаны два целых числа из диапазона $[-1000; +1000]$.

На **выход** нужно подать одно целое число - сумму чисел.

Sample Input:

1 2

Sample Output:

3

Примеры оформления кода для задачи:

C#

```
using System;
using System.Linq;

class Program
{
    static void Main()
    {
        Console.WriteLine(Console.ReadLine().Split().Select(x=>int.Parse(x)).Sum());

        Console.ReadLine(); // эту строку можно оставить
    }
}
```

Python

```
print(sum(map(int, input().split())))
```

Возможные ошибки

Если программа не прошла какой-то тест, участник получает поясняющее сообщение об ошибке:

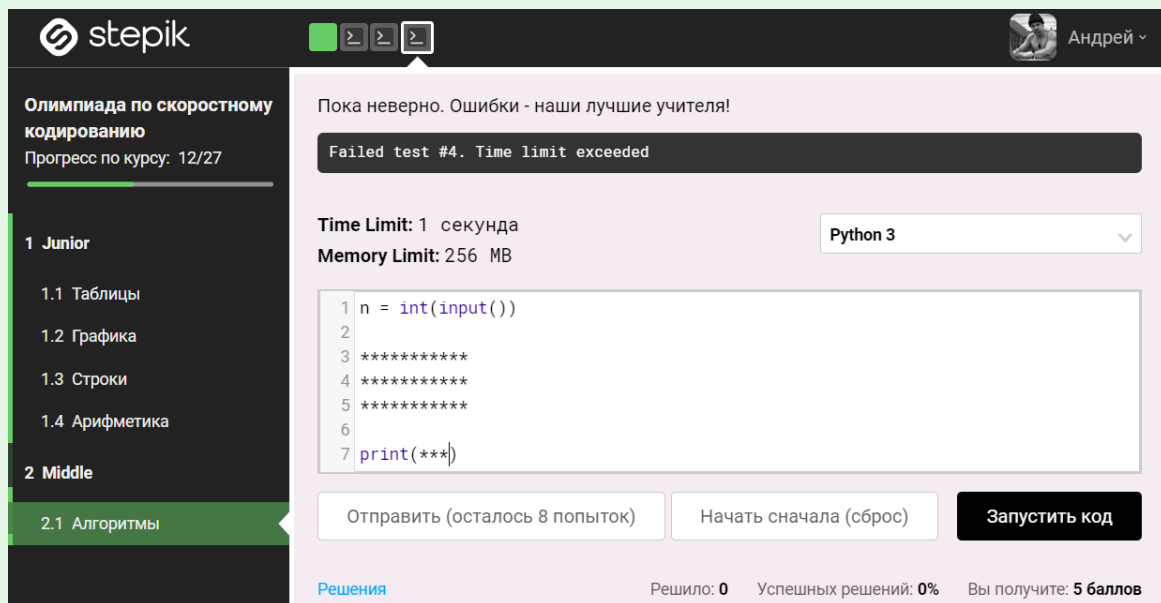
Compilation error — компиляция программы завершилась неудачно. Вероятно, программа содержит синтаксические или другие ошибки.

Wrong answer — программа выдала неправильный ответ на тест.

Time limit exceeded — программа работала дольше установленного Time Limit.

Runtime error — программа завершилась аварийно с ненулевым кодом возврата. Причиной может быть превышение установленного Memory Limit или ошибка в программе, которая привела к возникновению исключительной ситуации во время исполнения (например, деление на ноль, выход за пределы массива).

Для некоторых задач условия будут составлены так, что «наивное» решение может не уложиться в отведённое время или занять слишком много места в памяти, тогда вы получите примерно вот такое сообщение об ошибке:



The screenshot shows the Stepik interface for a programming problem. The sidebar on the left indicates the user is in the 'Олимпиада по скоростному кодированию' (Speed Coding Olympiad) course, with 12/27 progress. The main area displays a 'Failed test #4. Time limit exceeded' error message. Below the message, the 'Time Limit' is 1 second and the 'Memory Limit' is 256 MB. The code editor shows a Python script that reads an integer 'n' and prints a line of asterisks. The submission buttons include 'Отправить (осталось 8 попыток)', 'Начать сначала (сброс)', and 'Запустить код'. At the bottom, it shows 'Решения' (Solutions) and statistics: 'Решило: 0', 'Успешных решений: 0%', and 'Вы получите: 5 баллов'.

Обращайте внимание на уточнения в задаче по поводу формата вывода, например, в задаче «**Отражение матрицы**» указано, что

Не допускаются лишние пробелы вне квадрата.

Это буквально означает, что при таком выводе:

Sample Output 1:

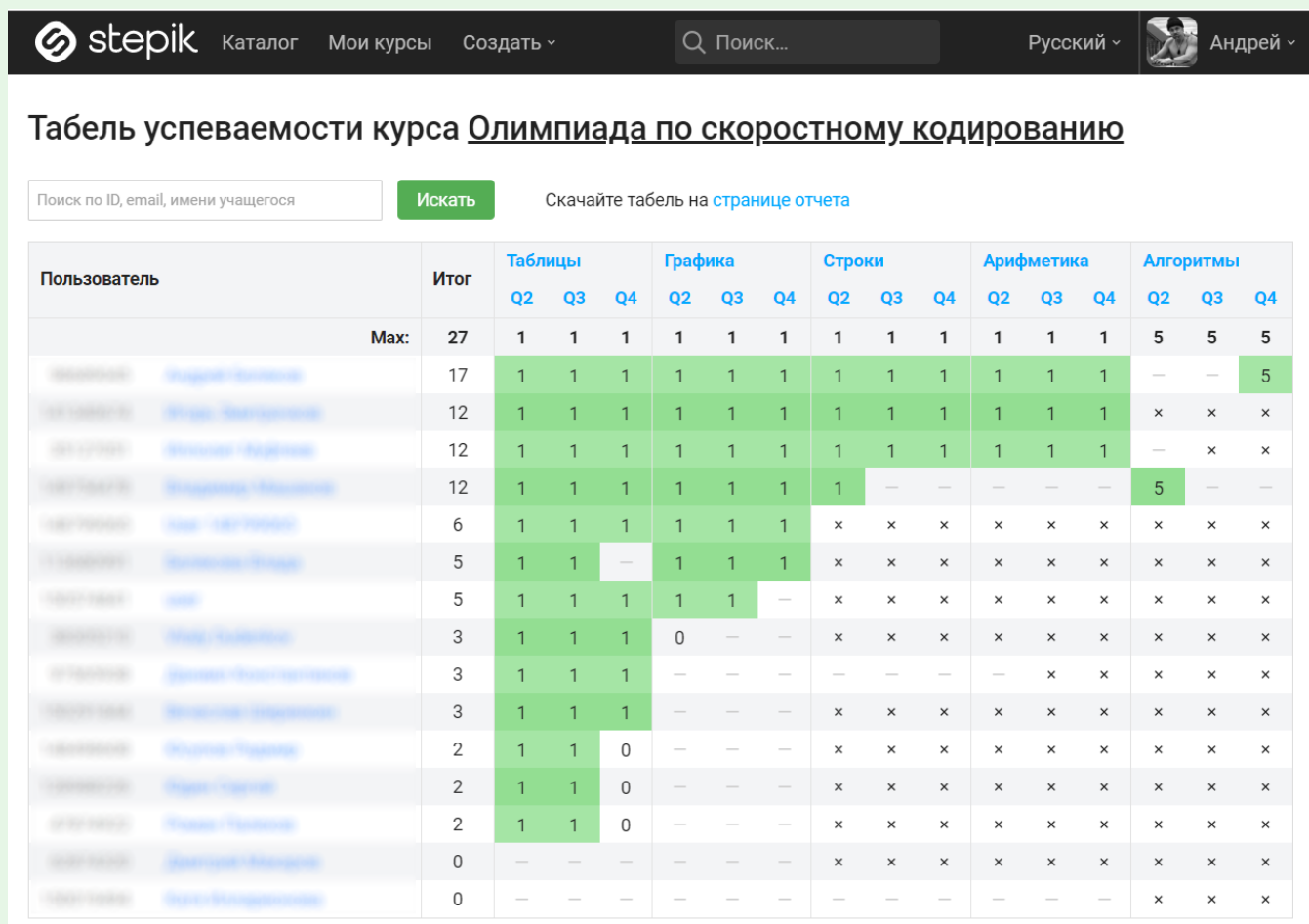
```
1 3
2 4
```

правее цифры 3 в первой строке вывода и правее цифры 4 во второй строке не нужно выводить пробел, иначе решение не будет засчитано.

Распределение мест

Распределение мест производится на основании заработанных баллов. Каждая задача сколько-то стоит (от 1 до 15 баллов) – это указано на экране задачи. В большинстве задач количество попыток будет ограничено – 5 (для простых) или 10 (для сложных). Более сложные задачи будут стоить заметно дороже (примерно в 3 – 5 раз дороже). При одинаковом количестве набранных баллов приоритет за тем, кто набрал баллы раньше (быстрее).

Пример распределения мест по итогам соревнования:



The screenshot shows the Stepik website interface. At the top, there is a navigation bar with the Stepik logo, 'Каталог', 'Мои курсы', 'Создать', a search bar, 'Русский', and a user profile for 'Андрей'. Below the navigation bar, the page title is 'Табель успеваемости курса Олимпиада по скоростному кодированию'. There is a search input field with the placeholder 'Поиск по ID, email, имени учащегося' and a green 'Искать' button. A link says 'Скачайте таблицу на [странице отчета](#)'. The main content is a table with the following structure:

Пользователь	Итог	Таблицы			Графика			Строки			Арифметика			Алгоритмы		
		Q2	Q3	Q4	Q2	Q3	Q4	Q2	Q3	Q4	Q2	Q3	Q4	Q2	Q3	Q4
Мах:	27	1	1	1	1	1	1	1	1	1	1	1	1	5	5	5
Андрей Андреевич	17	1	1	1	1	1	1	1	1	1	1	1	1	–	–	5
Александр Александрович	12	1	1	1	1	1	1	1	1	1	1	1	1	x	x	x
Александр Александрович	12	1	1	1	1	1	1	1	1	1	1	1	1	–	x	x
Александр Александрович	12	1	1	1	1	1	1	1	–	–	–	–	–	5	–	–
Александр Александрович	6	1	1	1	1	1	1	x	x	x	x	x	x	x	x	x
Александр Александрович	5	1	1	–	1	1	1	x	x	x	x	x	x	x	x	x
Александр Александрович	5	1	1	1	1	1	–	x	x	x	x	x	x	x	x	x
Александр Александрович	3	1	1	1	0	–	–	x	x	x	x	x	x	x	x	x
Александр Александрович	3	1	1	1	–	–	–	–	–	–	–	x	x	x	x	x
Александр Александрович	3	1	1	1	–	–	–	x	x	x	x	x	x	x	x	x
Александр Александрович	2	1	1	0	–	–	–	x	x	x	x	x	x	x	x	x
Александр Александрович	2	1	1	0	–	–	–	x	x	x	x	x	x	x	x	x
Александр Александрович	2	1	1	0	–	–	–	x	x	x	x	x	x	x	x	x
Александр Александрович	0	–	–	–	–	–	–	x	x	x	x	x	x	x	x	x
Александр Александрович	0	–	–	–	–	–	–	–	–	–	–	–	–	x	x	x

В данном соревновании задачи были стоимостью как 1 балл, так и 5 баллов. Места со 2-ое по 4-ое набрали одинаковое количество баллов, но распределены по затраченному времени. Количество неуспешных попыток не играет роли – штрафов не накладывается, важно уложиться в количество разрешённых попыток.

Технические особенности:

Для компиляции и выполнения решений используется следующее ПО и настройки:	Список идентификаторов и соответствующих им языков программирования, на которых вы можете подавать свои решения:
<p>ASM32 (gcc 6.3.0) Compile command: <code>gcc -pipe -m32 -o main</code></p> <p>ASM64 (gcc 6.3.0) Compile command: <code>gcc -pipe -m64 -o main</code></p> <p>C (gcc 6.3.0, C99 Standard) Compile command: <code>gcc -std=c99 -pipe -O2 -static -o main</code></p> <p>C# (Mono 5.0.0) Compile command: <code>mcs -optimize+</code></p> <p>C++ (g++ 6.3.0) Compile command: <code>g++ -pipe -O2 -static -o main</code></p> <p>C++11 (g++ 6.3.0) Compile command: <code>g++ -std=c++11 -pipe -O2 -static -o main</code></p> <p>Clojure (Clojure 1.9.0 using OpenJDK 8) Run command: <code>java -Xmx{memory limit} -Xss8m -cp clojure.jar clojure.main</code></p> <p>Go (1.8.1) Compile command: <code>go build -ldflags "-extldflags -static"</code></p> <p>Haskell Platform (ghc 7.8.3, 7.10.2, 8.0.1) Compile command: <code>ghc -O -static -optl-static -optl-pthread -o main</code></p> <p>Java (OpenJDK 7u181) Compile command: <code>javac</code> Run command: <code>java -Xmx{memory limit} -Xss8m MainName</code>. Where <i>MainName</i> is the auto-discovered main class that defines public static void main method.</p> <p>Java 8 (OpenJDK 8u181) Compile command: <code>javac</code> Run command: <code>java -Xmx{memory limit} -Xss8m MainName</code>. Where <i>MainName</i> is the auto-discovered main class that defines public static void main method.</p> <p>Java 9 (Oracle Java SE 9.0.1) Compile command: <code>javac</code> Run command: <code>java -Xmx{memory limit} -Xss8m MainName</code>. Where <i>MainName</i> is the auto-discovered main class that defines public static void main method.</p> <p>Java 11 (OpenJDK 11.0.1) Compile command: <code>javac</code> Run command: <code>java -Xmx{memory limit} -Xss8m MainName</code>. Where <i>MainName</i> is the auto-discovered main class that defines public static void main method.</p> <p>JavaScript (Node.js v10.13.0)</p>	<p><code>::asm32</code> — ASM32</p> <p><code>::asm64</code> — ASM64</p> <p><code>::c</code> — C</p> <p><code>::c++</code> — C++</p> <p><code>::c++11</code> — C++11</p> <p><code>::clojure</code> — Clojure</p> <p><code>::go</code> — Go</p> <p><code>::haskell</code> — Haskell (ghc 7.8)</p> <p><code>::haskell 7.10</code> — Haskell (ghc 7.10)</p> <p><code>::haskell 8.0</code> — Haskell (ghc 8.0)</p> <p><code>::java</code> — Java 7</p> <p><code>::java11</code> — Java 11</p> <p><code>::java8</code> — Java 8</p> <p><code>::java9</code> — Java 9</p> <p><code>::javascript</code> — JavaScript</p> <p><code>::julia</code> — Julia</p> <p><code>::kotlin</code> — Kotlin</p> <p><code>::mono c#</code> — C#</p> <p><code>::octave</code> — Octave</p> <p><code>::pascalabc</code> — PascalABC.NET</p> <p><code>::perl</code> — Perl 5</p> <p><code>::php</code> — PHP</p> <p><code>::python3</code> — Python 3</p> <p><code>::r</code> — R</p> <p><code>::ruby</code> — Ruby</p> <p><code>::rust</code> — Rust</p> <p><code>::scala</code> — Scala</p> <p><code>::shell</code> — Shell</p> <p><code>::swift</code> — Swift</p>

Run command: `node --max-old-space-size={memory limit}`

Julia (1.0.0)

Kotlin (Kotlin 1.3.11 using OpenJDK 11)

Compile command: `kotlinc`

Run command: `kotlin -J-Xmx{memory limit} -J-Xss8m`

MainKt

Octave (GNU Octave 3.8.2)

Run command: `octave --quiet --no-window-system --no-history --no-init-file --no-site-file --norc`

PascalABC.NET (3.2 using Mono 5.0.0)

Perl (5.24.3)

PHP (7.2.6)

Run command: `php -nf`

Python 3 (CPython 3.6.2)

R (3.3.3)

Run command: `R --vanilla --slave -f`

Ruby (CRuby 2.5.3)

Rust (rustc 1.31.0)

Compile command: `rustc -O -o main`

Scala (Scala 2.12.7 using OpenJDK 8u181)

Compile command: `scalac`

Run command: `scala -J-Xmx{memory limit} -J-Xss8m`

MainName. Where MainName is the auto-discovered main class that defines main method.

Shell (GNU bash 4.2.37)

Run command: `bash --noprofile --norc`

Swift (5.0.1 running on Ubuntu 18.04)

Compile command: `swiftc main.swift -o main`