

Инструкция по работе с автоматической тестирующей системой

Версия от 22 октября 2018 г.

На олимпиадах по информатике для проверки решений используются автоматические тестирующие системы. Для сдачи заданий в тестирующую систему необходимо понимать принципы их работы, изложенные в этой инструкции. Данная инструкция разработана для школьного, муниципального и регионального этапов всероссийской олимпиады школьников по информатике в г.Москве, но может быть использована и на других олимпиадах. В инструкции описаны особенности задач по информатике, приведены примеры оформления решений, описан интерфейс тестирующей системы ejudge, которая используется в г. Москве на этапах всероссийской олимпиады школьников по информатике.

Сначала описаны задачи по программированию, затем — задачи с вводом ответа, используемые на олимпиаде для 7-8 классов.

Оглавление

Как выглядит олимпиадная задача по информатике.....	2
Как выглядит решение олимпиадной задачи.....	3
Пример решения на языке Паскаль.....	4
Пример решения на языке Python версии 3.....	4
Пример решения на языке C.....	4
Пример решения на языке C++.....	4
Пример решения на языке C#.....	5
Пример решения на языке Java.....	5
Пример решения на языке Free Basic (аналог qbasic).....	5
Пример решения на языке Mono Visual Basic.....	5
Пример решения на языке Кумир.....	5
Пример решения на языке Perl.....	6
Пример решения на языке Ruby.....	6
Пример решения на языке PHP.....	6
Типичные ошибки при использовании тестирующей системы.....	6
Особенности использования разных языков программирования.....	7
Особенности использования языка Паскаль.....	7
Особенности использования языков Си и C++.....	8
Особенности использования языка Python.....	8
Особенности использования языка Бейсик.....	9
Особенности использования алгоритмического языка (Кумир).....	9
Особенности использования языка PHP.....	9
Интерфейс тестирующей системы.....	9
Сдача задач на проверку.....	11
Просмотр протокола проверки.....	12
Просмотр итога по всем задачам.....	15
Окончание олимпиады.....	16
Возможные результаты проверки на одном тесте.....	16
ОК.....	16
Неправильный ответ.....	16
Неправильный формат вывода.....	16
Ошибка выполнения.....	17
Превышено максимальное время работы.....	17
Превышен лимит по памяти.....	17

Как выглядит олимпиадная задача по информатике

Рассмотрим задачу «Конец K -го урока» с муниципального этапа олимпиады в 2009 году.

В школе продолжительность каждого урока 45 минут, а перемены между уроками – всего 5 минут. Первый урок начинается ровно в 8 часов утра. Напишите программу, отвечающую на вопрос «во сколько в этой школе заканчивается K -й урок?»

Программа получает на вход одно целое положительное число K , не превышающее 15.

Программа должна вывести время окончания K -го урока: сначала часы, потом минуты, разделяя их пробелом.

Пример входных и выходных данных

Ввод	Вывод
1	8 45
6	12 55

Условие олимпиадной задачи состоит из нескольких частей. Сначала написана «легенда» — условие задачи. Легенда может быть довольно длинной и на первый взгляд иметь довольно отдаленное отношение к задаче. Умение по легенде составить модель задачи и придумать алгоритм ее решения является необходимым навыком на олимпиадах по информатике.

На серьезных олимпиадах (например, на региональном этапе) перед легендой идут ограничения, в которые должно укладываться решение задачи — ограничение по времени работы программы на одном тесте (как правило 1-2 секунды) и ограничение по используемой памяти (как правило 64-256 мегабайт). Также в начале условия записываются имена файлов, которые используются для хранения входных и выходных данных, если олимпиада предполагает файловый ввод-вывод (на школьном и муниципальном этапах всероссийской олимпиады по информатике файловый ввод-вывод не используется).

Легенда отвечает на самый важный вопрос — «Что именно должна делать программа, которая является решением задачи?».

Далее идет часть, в которой описываются входные данные к задаче. В этом примере входными данными является одно целое число. В описании формата входных данных указываются **ограничения** на эти данные. В нашем случае сказано, что входное число является целым и положительным, и оно не превосходит 15. Ограничения на входные данные даются для того, чтобы участники олимпиады понимали, какие типы данных использовать для хранения чисел (например, если входное число может быть до 10^6 , то для его представления понадобится 32-битный целочисленный тип, например, тип `integer` для языка Паскаль не подойдет).

В следующей части описываются требования к тому, что должна вывести программа. В нашем случае программа выводит два числа — час и минуту, в которые заканчивается K -й урок.

Проверка программ на олимпиадах по информатике проводится с использованием автоматической тестирующей системы. В условиях этой задачи, например, программа должна давать правильный ответ для $K=1, 2, \dots, 15$. То есть если $K=1$, то ответом будет 8 часов 45 минут, если $K=2$ — 9 часов 35 минут, если $K=3$ — 10 часов 25 минут и т. д.

При этом вывод программы должен соответствовать в точности описанию формата выходных данных, то есть программа должна вывести только два целых числа. Для облегчения понимания, что должна выводить программа, в конце условия даются примеры того, что должна выводить программа при одном или нескольких вариантах входных данных.

Проверка заключается в том, что программа несколько раз запускается тестирующей системой и ей на вход подается заранее подготовленный жюри набор входных данных. Программа должна считать эти данные, решить задачу и вывести ответ. Ответ сверяется с правильным ответом, если ответ совпадает с правильным (или с одним из нескольких возможных правильных ответов, как это бывает в некоторых задачах), то тест засчитывается, как успешно пройденный. Если программа проходит все тесты (на всех тестах выдает правильный ответ), то задача считается верно решенной и получает полный балл, если не на всех тестах — то задача считается частично решенной и получает частичный балл в зависимости от количества пройденных тестов. Все это происходит в автоматическом режиме без участия человека, члены жюри только наблюдают за процессом проверки.

Как выглядит решение олимпиадной задачи

Решением задачи является программа на одном из допустимых языков программирования, которые поддерживаются тестирующей системой. Тестирующая система компилирует исходный код программы (если используется компилируемый язык, например, C++ или Паскаль), затем запускает программу и подает программе данные на вход и ожидает от программы вывода результата. Если не используются файлы, программа должна для считывания данных использовать стандартные функции ввода-вывода языка программирования, например, `read` и `write` в Паскале, `input` и `print` в бейсике, `scanf` и `printf` в C, `cin` и `cout` в C++, `input` и `print` в Python и т. д. В тестирующей системе стандартный ввод-вывод перенаправляется так, что программа будет считывать входные данные из заранее подготовленных жюри файлов с тестами, затем тестирующая система проверит правильность выведенного программой результата.

Решение не должно использовать никакие графические функции языка программирования и операционной системы, то есть нельзя осуществлять ввод-вывод через формы, окна диалогов и т. д., поскольку проверка происходит без участия человека, который будет вводить входные данные в формы. То есть создаваемое приложение должно быть простым «консольным» приложением, а не графическим приложением, использующим окна.

По этим же причинам не могут быть проверены программы, предполагающие взаимодействие через интернет или с использованием браузера. Например, программа на языке PHP предполагающая, что она будет запущена на web-сервере и будет получать данные через GET или POST-методы протокола http, не может быть проверена в автоматической системе. Аналогично не может быть проверена программа на JavaScript, для исполнения которой нужен web-браузер.

Прежде всего программа должна считать входные данные. При этом если программа будет выводить какие-то дополнительные сообщения, например, «Введите количество уроков», то данный текст также будет выведен программой и обработан тестирующей системой, то есть программа выведет не только два числа в ответе, но и дополнительное сообщение и не сможет быть проверена тестирующей системой. Поэтому решение не должно содержать команды вывода дополнительных сообщений вида `writeln('Введите количество уроков')`.

Не надо проверять входные данные на корректность (например, проверять, что вводится число, не превосходящее 15) — ограничения в условиях задачи означают, что во всех тестах, на которых будет проверяться ваша программа, будут выполнены данные ограничения. Инструкции вроде `if K > 15 then writeln('Введите правильное число уроков')` не

являются ошибкой (они никогда не сработают), но просто бессмысленны.

При выводе программы не надо выводить ничего лишнего, если требуется вывести два числа, то нужно вывести только два числа (разделив их при этом пробелом или переходом на новую строку в зависимости от условия задачи). Не нужно выводить никаких дополнительных сообщений вида «Ответ:», не нужно выводить слова «часы» или «минуты» — весь этот вывод также помешает тестирующей системе осуществить проверку решения.

В конце программы иногда ставят дополнительную задержку — например, ожидают ввода чего-нибудь при помощи функции `Readln` в Паскале или вызывают функцию `system("pause")` в языке C — это также не требуется, программа должна сразу же завершить свою работу после вывода результата, не дожидаясь какого-либо действия от человека.

Приведём примеры возможных решений этой задачи, подходящих для проверки в тестирующей системе:

Пример решения на языке Паскаль

```
var k, time: longint;
begin
    read(k);
    time := 45 * k + 5 * (k - 1);
    writeln(8 + time div 60, ' ', time mod 60)
end.
```

Пример решения на языке Python версии 3

```
k = int(input())
time = 45 * k + 5 * (k - 1)
print(8 + time // 60, time % 60)
```

Пример решения на языке C

```
#include<stdio.h>
int main()
{
    int k, time;
    scanf("%d", &k);
    time = 45 * k + 5 * (k - 1);
    printf("%d %d", 8 + time / 60, time % 60);
    return 0;
}
```

Пример решения на языке C++

```
#include<iostream>
using namespace std;
int main()
{
    int k, time;
    cin >> k;
    time = 45 * k + 5 * (k - 1);
    cout << 8 + time / 60 << " " << time % 60;
    return 0;
}
```

```
}
```

Пример решения на языке C#

```
using System;
using System.IO;

class Program
{
    static void Main()
    {
        int k = int.Parse(Console.ReadLine());
        int time = 45 * k + 5 * (k - 1);
        Console.WriteLine("{0} {1}", 8 + time / 60, time % 60);
    }
}
```

Пример решения на языке Java

```
import java.io.*;

public class Main
{
    public static void main(String[] args) throws Exception
    {
        DataInputStream in = new DataInputStream(System.in);
        int k, time;
        k = Integer.parseInt(in.readLine());
        time = 45 * k + 5 * (k - 1);
        System.out.println((8 + time / 60) + " " + (time % 60));
    }
}
```

Пример решения на языке Free Basic (аналог qbasic)

```
DIM k, time AS LONG
INPUT k
time = 45 * k + 5 * (k - 1)
print 8 + time \ 60, time MOD 60
```

Пример решения на языке Mono Visual Basic

```
Module ProgramA
Sub Main()
DIM k, time AS INTEGER
k = CInt(Console.ReadLine())
time = 45 * k + 5 * (k - 1)
Console.WriteLine(CStr(8 + time \ 60) + " " + CStr(time MOD 60))
End Sub
End Module
```

Пример решения на языке Кумир

```
алг Задача А
нач
```

```
цел k, time
ввод k
time := 45 * k + 5 * (k - 1)
вывод 8 + div(time, 60), ' ', mod(time, 60)
кОН
```

Пример решения на языке Perl

```
my $k = <>;
my $time = 45 * $k + 5 * ($k - 1);
print 8 + int($time / 60), " ", $time % 60, "\n";
```

Пример решения на языке Ruby

```
k = gets.to_i
time = 45 * k + 5 * (k - 1)
print 8 + time / 60, " ", time % 60, "\n"
```

Пример решения на языке PHP

```
<?php
$k = fgets(STDIN);
$time = 45 * $k + 5 * ($k - 1);
print 8 + floor($time / 60) . " " . $time % 60 . "\n";
?>
```

Типичные ошибки при использовании тестирующей системы

В условии задачи приведены примеры входных и выходных данных, прежде всего решение проверяется на примерах входных и выходных данных из условия. В правилах олимпиады может быть указано, что решение будет проверяться, только если оно выдаёт правильный ответ на примерах из условия (например, такое требование есть на школьном и муниципальном этапе в г. Москве). При этом доступен полный протокол проверки на примерах тестов из условия, то есть в тестирующей системе будет показано, что вывела программа на тестах из условия.

После сдачи задания в тестирующую систему необходимо убедиться в том, что решение хотя бы проходит тесты из условия (а ещё лучше — если оно набирает много баллов). Если решение не проходит тесты из условия, то необходимо просмотреть протокол проверки, как правило, в протоколе содержится информация о причинах непрохождения программой тестов.

В большинстве случаев ошибки связанные с тем, что программа не может быть проверена тестирующей системой, имеют одну из следующих причин:

1. Программа не является консольным приложением, либо используются какие-то нестандартные возможности компилятора, привязанные к конкретной операционной системе, например, использование функции `clrscr` в Паскале, прекомпилированных заголовочных файлов в Visual C++ и т. д.
2. В программах на Паскале используется модуль `crt`.
3. Программа выводит дополнительные сообщения, «улучшающие» интерфейс, например, «Введите количество уроков».

4. Программа выводит дополнительные сообщения, не предусмотренные форматом выходных данных, например, «часы», «минуты», «ответ».
5. Программа содержит «задержку» после окончания работы, то есть ждет от пользователя нажатия на какую-либо клавишу, ввода строки или иных действий.
6. Входные данные считываются не так как в условии задачи. Наиболее распространенная ошибка — в описании входных данных говорится, что входные числа находятся в разных строках, а пользователь вводит их в одной строке.
7. Программа должна вывести целое число, но из-за того, что в программе вместо целочисленного типа используется действительный тип данных (например, тип `real` в языке Паскаль), программа выводит ответ в виде действительного числа, запись которого содержит точку или символ «e».

Ниже указаны особенности конкретных языков программирования и приведены примеры правильных программ на данных языках программирования.

Особенности использования разных языков программирования

Особенности использования языка Паскаль

Язык программирования Паскаль представлен в тестирующей системе компиляторами `Free Pascal` и `PascalABC.Net`.

В программах на языке Паскаль запрещается использовать модуль `crt` (даже подключение этого модуля может сделать невозможным проверку программы).

Одной из типичных ошибок ввода-вывода при использовании языка Паскаль является неправильное использование функции `ReadLn`, которая после считывания данных считывает конец строки (и пропускает все данные, находящиеся после считанных данных и до конца строки). Как правило это возникает в случае, когда программа получает на вход два или более чисел, записанных в одной строке через пробел. В этом случае нельзя читать данные при помощи `ReadLn(a); ReadLn(b)`, так как такое использование функций ввода означает, что после первого числа должен быть конец строки, и следующее число записано в новой строке. Правильным чтением данных будет `ReadLn(a, b)` или `Read(a); Read(b)`.

Если программа должна считывать только числа, то на языке Паскаль для чтения всегда лучше использовать функцию `Read` вместо `ReadLn`.

Особенности компилятора `Free Pascal`

Компилятор `Free Pascal` по умолчанию работает в 16-битном режиме, в частности, размер переменной `integer` составляет 2 байта (16 бит) и переменная такого типа может принимать значения от -32768 до 32767. Для хранения 32-битных целых чисел во `Free Pascal` следует использовать тип `longint`. Длина текстовой строки во `Free Pascal` ограничена 255 символами.

Данные параметры можно менять внутри программы при помощи директив компилятора, описанных в документации на язык `Free Pascal`.

В программах на `Free Pascal` нельзя использовать модуль `crt` и функции из этого модуля, например, `ClrScr`.

Особенности компилятора PascalABC.NET

В программах на PascalABC.NET нельзя использовать модуль `crt` и функции из этого модуля, например, `ClrScr`.

Особенности использования языков Си и C++

В тестирующей системе доступен компилятор GNU C/C++. Тестирующая система `ejudge` работает под управлением операционной системы Linux, поэтому компилятор Visual C/C++ тестирующей системой не поддерживается.

Для разработки программ можно использовать среду Visual C++, но проверять решения нужно под компилятором GNU C/C++. При создании проекта в Visual Studio (меню File — New — Project) необходимо выбрать «Win32 Console Application». При создании проекта необходимо отключить использование прекомпилированных заголовочных файлов (в окне диалога «Application Settings» убрать галочку «Additional options: Precompiled headers»). В готовом файле с программой не должно быть строки

```
#include "stdafx.h"
```

(наличие этой строки означает, что не были отключены прекомпилированные заголовочные файлы). Также описание функции `main` необходимо изменить на

```
int main()
```

вместо `int _tmain(int argc, _TCHAR* argv[])`. (если среда Visual Studio создала такой шаблон для программы).

Программа на языках C или C++ должна заканчиваться с кодом возврата 0 (`return 0`), ненулевой код возврата может быть воспринят тестирующей системой, как ошибка в работе программы. Функция `main` должна возвращать значение типа `int` (а не `void`).

Программа не должна содержать задержки после вывода результата типа вызова функции `system("pause")`.

В программе необходимо явно подключать заголовочные файлы, т. к. компилятор в тестирующей системе не подключает автоматически ни одного заголовочного файла с функциями стандартной библиотеки. То есть в программах на языке C должно быть написано

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

В программах на языке C++ должно быть написано

```
#include <iostream>
```

Особенности использования языка Python

В настоящий момент существует две версии языка Python: вторая (например, 2.7.15) и третья (например, 3.7.1). Вторая и третья версии имеют достаточно существенные различия. В тестирующей системе также представлены две версии, поэтому выбирать версию языка Python нужно внимательно. Если вы не знаете, какую версию выбрать, то скорее всего нужно выбирать версию 3 (версия 2 уже практически не используется).

«Минорные» версии языка Python (например, 3.4.5, 3.6.7, 3.7.1) имеют незначительные различия, проблем с различием минорных версий, скорее всего, не возникнет.

В программах на языке Python необходимо обращать внимание на то, как подаются числа на вход программе. Если в строке записано одно число, то его нужно считывать так:

```
a = int(input())
```

Если в строке записано несколько (например, три) числа через пробел, то их можно считать следующим способом:

```
a, b, c = map(int, input().split())
```

или

```
a, b, c = [int(x) for x in input().split()]
```

Особенности использования языка Бейсик

Региональная методическая комиссия всероссийской олимпиады школьников по информатике в г. Москве не рекомендует использовать язык программирования Бейсик на олимпиадах и для обучения программированию!

Компилятор Qbasic не поддерживается в тестирующей системе, вместо него используется компилятор Free BASIC (<http://www.freebasic.net/>), который будет запускаться в режиме совместимости с Qbasic (с ключом компиляции `-lang qb`). В среде разработки QuickBasic 4.5 при сохранении файла обязательно нужно выбрать формат «Text — Readable by Other Programs», в противном случае среда QuickBasic сохраняет файл в двоичном формате, который не может быть считан другими компиляторами.

Компилятор Visual Basic в тестирующей системе доступен в варианте Mono Visual Basic, при этом для работы программы необходимо создавать консольное приложение, а не графическое приложение. Программа должна читать данные со стандартного ввода, выводить результат на стандартный вывод. Компилятор Mono Visual Basic по своим возможностям соответствует Microsoft Visual Basic 2005.

Особенности использования алгоритмического языка (Кумир)

Программа, написанная на русском алгоритмическом языке, должна быть обязательно подготовлена (написана) в среде разработки Кумир. Программы, написанные с использованием других текстовых редакторов (например, Блокнот, Word и т. д.) не могут быть проверены тестирующей системой.

Это связано с тем, что среда разработки Кумир сохраняет файлы в малораспространенной кодировке UTF-16, поэтому проверяющая система предполагает, что файлы сохранены именно с использованием кодировки UTF-16. Файлы, сохраненные с использованием другой кодировки, тестирующей системой проверены быть не могут.

Особенности использования языка PHP

Программы на языке PHP могут быть проверены, только если они написаны для консольного режима CLI (command line interface). В этом режиме программа считывает данные со стандартного ввода и выводит результат на стандартный вывод. Программы на языке PHP написанные для web-сервера (получающие данные и выводящие результат по http-протоколу) не могут быть проверены в тестирующей системе. Участникам олимпиады, желающим использовать язык PHP необходимо заранее научиться реализовывать и запускать программы в режиме CLI.

Интерфейс тестирующей системы

Во время участники олимпиады сдают задачи на проверку в тестирующую систему. Для этого необходимо войти в тестирующую систему, указав свой логин и пароль. Основная

страница тестирующей системы во время проведения олимпиады имеет следующий вид.

Пользователь [Тренировочный тур для школьного этапа 2018 года в Москве, 9-11 классы]: Сдать решение

Настройки Регистрация Инфо Закончить виртуальный турнир Итог Посылки Выйти из системы [test_user]

21:52:03 / RUNNING / Остается: 11:49:14

1 2 3 4 5

Сдать решение задачи 1

Полный балл: 100
Ограничение времени: 1 с
Ограничение памяти: 256M

Язык: pasabc-linux - Pascal ABC.NET v3.0.1573

Файл: Обзор... Файл не выбран.

Отправить! Отправить!

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
10285	2018/10/21 21:51:16	105	1	pasabc-linux	OK	11	100	Просмотр	Просмотр
10276	2018/10/21 21:46:05	95	1	pasabc-linux	Ошибка компиляции	Неизв.	Неизв.	Просмотр	Просмотр
10273	2018/10/21 21:45:06	105	1	pasabc-linux	Неполное решение	0	0	Просмотр	Просмотр
10272	2018/10/21 21:43:49	112	1	pasabc-linux	Неполное решение	8	70	Просмотр	Просмотр

1 2 3 4 5

Эта страница используется для просмотра сданных решений по задачам и сдачи новых решений. В верхней части экрана содержится меню ссылок тестирующей системы.

- «Настройки» — выбор языка интерфейса тестирующей системы.
- «Регистрация» — редактирование регистрационной информации (имя, школа, класс)
- «Инфо» — просмотр информации о состоянии сервера и турнире.
- «Закончить виртуальный турнир» — окончание олимпиады и получение результатов проверки (для задач с записью ответа для 7-8 классов).
- «Итог» — сводная таблица результатов по всем задачам.
- «Посылки» — список отправленных на проверку решений по всем задачам.

На муниципальном этапе также в меню будут содержаться ссылки «Отправить вопрос» для отправки вопроса по условиям задачи или для иных сообщений жюри «Сообщения» — страница, на которой можно увидеть отправленные сообщения и полученные ответы на них.

Ниже находится статусная строка с информацией о текущем состоянии турнира.

Ниже находится меню вкладок с номерами задач: «1», «2», «3», «4», «5». Щёлкните по номеру задачи для сдачи этой задачи.

Ниже приведена информация по задаче: балл за задачу, ограничение по времени работы программы (1 с), ограничение по используемой памяти (256 Мбайт).

Ниже содержится форма отправки решения задачи.

Ниже содержится таблица со списком ранее сданных решений по этой задаче.

Сдача задач на проверку

Для сдачи задачи необходимо выбрать вкладку с номером задачи («1», «2» и т. д.), выбрать язык программирования из списка, выбрать файл на компьютере и нажать на кнопку «Отправить».

Выбранный файл отправляется на проверку в тестирующий сервер и решение появляется в списке «Предыдущие решения этой задачи».

Тестирование занимает некоторое время, обычно — не более минуты. При большой нагрузке на сервер проверка может занимать несколько минут. Во время тестирования в поле «Результат» для данного решения будет написано «Компилируется...» или «Выполняется...», которое после окончания проверки сменится на результат ОК, «Ошибка компиляции» или «Неполное решение». Для обновления страницы в браузере нажмите клавишу F5 или Ctrl+R, или кнопку на панели инструментов браузера.

Список «Предыдущие решения этой задачи» может содержать:

- Номер решения — уникальный номер отправленного решения в данном турнире.
- Время отправки решения.
- Размер отправленного файла в байтах.
- Номер задачи, по которой сдано решение.
- Язык программирования, выбранный при сдаче решения.
- Результат проверки данного решения.
- Количество тестов, которое прошло это решение.
- Количество баллов, набранных данным решением.
- Ссылка на просмотр исходного кода решения (может отсутствовать во время олимпиады).
- Ссылка на просмотр протокола проверки данного решения.

Последнее сданное решение находится сверху таблицы. На приведённом выше снимке экрана по задаче 1 было сдано 4 решения — первое набрало 70 баллов, второе — не прошло ни одного теста и набрало 0 баллов, третье получило ошибку компиляции, наконец, четвёртое прошло все тесты и получило результат ОК и 100 баллов.

После окончания проверки в поле «Результат» может быть записано следующее.

«Ошибка компиляции» означает, что решение содержит синтаксические ошибки и не было скомпилировано тестирующей системой. Возможные причины для этого:

- Синтаксическая ошибка в программе.
- Неверно выбран язык программирования при сдаче задачи.
- Неверно выбран файл при сдаче задачи.
- Файл с решением задачи был изменён в среде разработки, но при этом изменения не были сохранены перед сдачей.
- Есть различия в используемых версиях компилятора в тестирующей системе и участником олимпиады.

Во всех случаях необходимо нажать на ссылку «Просмотр» в столбце «Просмотреть протокол». Протокол проверки будет содержать вывод компилятора, с указанием имеющихся ошибок.



22:43:50 / RUNNING / Остается: 10:57:27

Ошибка компиляции

```
Loading core...
Подключен парсер PascalABC.NET Language Parser v1.2
Подключен парсер Documentation Comments Tag Parser v0.9
PascalABCCompiler.Core v3.4.2.1806
Copyright (c) 2005-2018 by Ivan Bondarev, Stanislav Mihalkovich
OK 132.413ms
Connected parsers: PascalABC.NET (*.pas);
Connected conversions: Optimizer;
Compiling assembly 010276.pas...
[0][3,10] 010276.pas: Undefined name reference 'k'
```

На изображении выше указано, что в программе на языке PascalABC.NET в строке 3 программы используется необъявленная переменная «k»:

```
[0][3,10] 010276.pas: Undefined name reference 'k'
```

Необходимо исправить ошибки в программе и сдать программу заново.

Если в столбце «Результат» написано «Частичное решение», то это означает, что программа была скомпилирована, но прошла не все тесты. В этом случае в таблице указана информация о числе пройденных тестов и количестве набранных баллов, а протокол проверки содержит более подробную информацию.

Результат «ОК» означает, что решение прошло все тесты и получило полный балл.

Просмотр протокола проверки

Протокол проверки содержит следующую информацию.

Неполное решение

Всего тестов: 11, пройдено: 8, не пройдено: 3.
Получено баллов: 70 (из 100).

N	Результат	Время (с)	Баллы
1	OK	0.249	0 (0)
2	OK	0.250	10 (10)
3	OK	0.252	10 (10)
4	OK	0.251	10 (10)
5	OK	0.256	10 (10)
6	OK	0.252	10 (10)
7	Неправильный ответ	0.249	0 (10)
8	Неправильный ответ	0.252	0 (10)
9	Неправильный ответ	0.252	0 (10)
10	OK	0.249	10 (10)
11	OK	0.251	10 (10)

===== Тест #1 =====

--- Входные данные: размер 9 ---

33

5

100

--- Результат работы: размер 2 ---

2

--- Правильный ответ: размер 2 ---

2

--- Поток ошибок: размер 0 ---

--- Вывод проверяющей программы: размер 20 ---

ok 1 number(s): "2"

Вверху идет таблица в которой для каждого теста указан номер теста, результат запуска программы на этот тесте («OK», «Неправильный ответ», «Превышено максимальное время работы» и т. д.), время работы программы на этом тесте в секундах, количество баллов, которое получила программа за прохождение этого теста (в скобках — максимальное количество баллов за этот тест). На изображении ниже приведён пример решения, которое было проверено на 11 тестах. Тест № 1 — пример из условия задачи, он оценивается в 0 баллов. Следующие 10 тестов оцениваются в 10 баллов каждый. Программа прошла все тесты, кроме трёх (№№ 7, 8, 9), поэтому программа оценивается в 70 баллов.

Поскольку тест № 1 является примером из условия задачи, то ниже приведён полный протокол проверки на тесте № 1. Это протокол содержит разделы:

- «Входные данные» — то, что было подано на вход программе.
- «Результат работы» — то, что вывела программа на этих входных данных при запуске в тестирующей системе.
- «Правильный ответ» — что является правильным ответом для данного набора входных данных.
- «Поток ошибок» — что программа вывела в стандартный поток сообщений об ошибках (stderr, например, в программах на языке C++ в этот поток вывод осуществляется при помощи `cerr << ...`), что может использоваться для вывода отладочной информации. Тестирующая система игнорирует вывод программы в стандартный поток сообщений об ошибках.

- «Вывод проверяющей программы» — здесь может содержаться полезная информация, которую вывела программа, используемая для проверки ответа на правильность.

В данном примере проверяющая программа вывела сообщение о том, что в ответе указано 1 число, это число «2», и это правильный ответ.

Приведём пример протокола, в котором решение не проходит пример из условия.

Неполное решение

Всего тестов: 11, пройдено: 0, не пройдено: 11.
Получено баллов: 0 (из 100).

Комментарии оценивания:

Тестирование на тестах 2-11 не выполнялось, так как не пройдена одна из требуемых групп 0.

N	Результат	Время (с)
1	Неправильный формат вывода	0.257
2	Пропущен	0.000
3	Пропущен	0.000
4	Пропущен	0.000
5	Пропущен	0.000
6	Пропущен	0.000
7	Пропущен	0.000
8	Пропущен	0.000
9	Пропущен	0.000
10	Пропущен	0.000
11	Пропущен	0.000

===== Тест #1 =====

--- Входные данные: размер 9 ---

33

5

100

--- Результат работы: размер 17 ---

2.88600288600289

--- Правильный ответ: размер 2 ---

2

--- Поток ошибок: размер 0 ---

--- Вывод проверяющей программы: размер 67 ---

wrong output format Expected integer, but "2.88600288600289" found

В этом примере программа вывела в качестве ответа действительное число 2.88600288600289, а по условию задачи программа должна вывести целое число, поэтому результат проверки — «неправильный формат вывода». Отметим, что если в условии задачи содержится требование о том, что программа должна вывести целое число, то вывод «2.0», то есть вывод действительного числа, равного целому числу 2, также будет считаться

неправильным. Для тестов 2-11 указан результат «Пропущен». Это означает, что тестирование на этих тестах не выполнялось, потому что программа не прошла тест из условия задачи.

Приведём пример протокола проверки решения, проходящего все тесты.

OK

Всего тестов: 11, пройдено: 11, не пройдено: 0.

Получено баллов: 100 (из 100).

N	Результат	Время (с)	Баллы
1	OK	0.251	0 (0)
2	OK	0.247	10 (10)
3	OK	0.248	10 (10)
4	OK	0.250	10 (10)
5	OK	0.250	10 (10)
6	OK	0.293	10 (10)
7	OK	0.262	10 (10)
8	OK	0.290	10 (10)
9	OK	0.248	10 (10)
10	OK	0.248	10 (10)
11	OK	0.251	10 (10)

===== Тест #1 =====

--- Входные данные: размер 9 ---

33

5

100

--- Результат работы: размер 2 ---

2

--- Правильный ответ: размер 2 ---

2

--- Поток ошибок: размер 0 ---

--- Вывод проверяющей программы: размер 20 ---

ok 1 number(s): "2"

Просмотр итога по всем задачам

Ссылка «Итог» отображает таблицу с результатом сдачи решений по всем задачам.

Итог по задачам				
Задача	Статус	Пройдено тестов	Баллы	Номер решения
1	OK	11	100	10285
2	OK	21	100	10373
3	OK	11	100	10375
4	Неполное решение	11	45	10378
5				

Суммарный балл: 345

В этом примере по первым трём задачам были сданы решения на 100 баллов, по

четвёртой задаче — решение на 45 баллов, по пятой задаче ничего не было сдано. По каждой задаче засчитывается наилучшее решение.

По ссылке «Посылки» можно просмотреть список всех сданных решений по всем задачам.

Отправленные решения (последние 15)									
Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
10378	2018/10/21 23:19:56	364	4	python3	Неполное решение	11	45	Посмотреть	Посмотреть
10375	2018/10/21 23:19:31	165	3	python3	OK	11	100	Посмотреть	Посмотреть
10373	2018/10/21 23:19:11	370	2	python3	OK	21	100	Посмотреть	Посмотреть
10328	2018/10/21 22:20:03	107	1	pasabc-linux	OK	11	100	Посмотреть	Посмотреть
10325	2018/10/21 22:19:18	106	1	pasabc-linux	OK	11	100	Посмотреть	Посмотреть
10285	2018/10/21 21:51:16	105	1	pasabc-linux	OK	11	100	Посмотреть	Посмотреть
10276	2018/10/21 21:46:05	95	1	pasabc-linux	Ошибка компиляции	Неизв.	Неизв.	Посмотреть	Посмотреть
10273	2018/10/21 21:45:06	105	1	pasabc-linux	Неполное решение	0	0	Посмотреть	Посмотреть
10272	2018/10/21 21:43:49	112	1	pasabc-linux	Неполное решение	8	70	Посмотреть	Посмотреть

[Посмотреть всё](#)

Окончание олимпиады

После окончания олимпиады, как правило, становятся доступны полные протоколы проверки, в которых приведены входные и выходные данные и правильные ответы на всех тестах.

Возможные результаты проверки на одном тесте

Перечислим возможные вердикты тестирующей системы для отдельного теста.

OK

Тест пройден, программа выдала правильный ответ на этом тесте.

Неправильный ответ

Программа выдала неправильный ответ на данном тесте, это означает, что программа содержит ошибку.

Неправильный формат вывода

То, что вывела программа, не соответствует описанию формата выходных данных, приведенному в условию задачи. Возможные причины для этого такие:

1. Программа не вывела ничего (это может быть, например, при наличии ошибки в программе или при ошибке в написании имени выходного файла, если используется файловый ввод-вывод).
2. Программа выводит лишние сообщения типа «Введите число» или «Ответ».
3. Программа должна вывести два числа, а вывела одно число, или три числа, или текст и т. д.
4. Программа должна вывести числа в одной строке через пробел, а вывела их в разных строках или наоборот.
5. Программа должна вывести целое число, а выводит действительное число, запись которого содержит десятичную точку, из-за того, что программа использует действительный тип данных (например, тип `real` в языке Паскаль), вместо целочисленного.

Ошибка выполнения

Программа совершила некорректную операцию во время тестирования. Возможные причины для этого.

1. Некорректное арифметическое действие или математическая операция, например, деление на ноль, извлечение корня из отрицательного числа, переполнение переменных.
2. Ошибки при работе с памятью — выход за границы массива, обращение к невалидным указателям в языке C или C++, переполнение стека, выделение слишком большого объема динамической памяти.
3. Программа неверно читает входные данные — например, программа получает на вход несколько чисел, записанных каждое в отдельной строке, а считывает программа так, как будто бы они находятся в одной строке через пробел.
4. В программе на языке Python и иных интерпретируемых языках — синтаксическая ошибка в программе или любая иная ошибка при исполнении программы. Например, использование переменной, которой ранее не было присвоено значение, приведёт к ошибке исполнения на языке Python.
5. Бесконечная (или очень большая) рекурсия.
6. В программе на языках C или C++ явно указан ненулевой код завершения программы.

Превышено максимальное время работы

Программа не закончила свою работу за время, отведенное на исполнение одного теста. Возможные причины для этого:

1. Алгоритмическая ошибка — программа попадает в «бесконечный цикл».
2. Неэффективное решение — программа работает слишком долго.
3. Ошибка в считывании данных, например, программа считывает со стандартного ввода два числа, между тем как в описании формата входных данных указано только одно число (тогда программа будет бесконечно долго ожидать ввода второго числа).
4. Программа явно ожидает от пользователя ввода чего-либо, нажатия на клавишу после решения задачи (для организации «задержки» программы).

Превышен лимит по памяти

Программа использовала больше оперативной памяти, чем это предусмотрено ограничениями в задаче. Возможные причины для этого:

1. Используются слишком большие массивы (или слишком много массивов) или иные структуры данных.
2. Бесконечная (или очень большая) рекурсия.
3. Некорректная работа с указателями в C или C++ также может диагностироваться, как «Превышен лимит по памяти».

Пропущен

Проверка на этом тесте не производилась. Например, задача не проходит тесты из условия — в этом случае проверка на других тестах не производится.

Задания с вводом ответа для 7-8 классов

На школьном и муниципальном этапе олимпиады для 7-8 классов в г. Москве предлагаются задания с вводом ответа. Для выполнения этих заданий не нужно уметь программировать, ответом на задание является текст, который вводится в однострочное или многострочное поле ввода. Рассмотрим пример такой задачи.

Дан список чисел:

3, 7, 1, 6, 2, 4, 8, 5.

Разрешается за одну операцию поменять местами два любых числа в этом списке. Например, если поменять местами числа 6 и 8, то получится список 3, 7, 1, 8, 2, 4, 6, 5.

Упорядочите этот список по возрастанию, то есть получите из него список 1, 2, 3, 4, 5, 6, 7, 8, используя минимальное число обменов.

Решение этой задачи нужно записать в виде последовательности обменов, каждый обмен записывается в одной строке. Один обмен записывается в виде двух различных чисел от 1 до 8, которые нужно поменять местами, записанных через пробел (пример: 6 8).

Для сдачи этой задачи необходимо ввести ответ в поле ввода, затем нажать на кнопку «Отправить!».

The screenshot shows a web interface for an online competition. At the top, a green status bar displays the time '09:51:56 / RUNNING / Остаётся: 11:58:50'. Below this is a navigation bar with buttons numbered 1 through 7, where button 2 is highlighted in blue. The main content area is titled 'Сдать решение задачи 2'. It features a text input field containing the following text:

```
1 3
2 7
4 6
5 7
7 8
```

At the bottom of the input field, there are two buttons labeled 'Отправить!' (Send!).

Условие задачи предполагает много возможных правильных ответов (нет единственно возможного правильного ответа), поэтому для проверки ответа также используется проверяющая программа. Для того, чтобы решение могло быть проверено, необходимо, чтобы оно соответствовало требованиям к формату записи ответа, описанному в условии задачи. В условии этой задачи написано, что в ответе должно быть несколько строк, в каждой строке записано два различных числа от 1 до 8 через пробел. Никаких других символов в

строке быть не должно.

После отправки ответа, осуществляется проверка на соответствие сданного ответа формату, описанному в условии задачи. Если форма записи ответа не соответствует условию задачи, то результатом проверки будет «Неправильный формат ответа». Например, запишем ответ, поставив вместо пробела между цифрами знак «-».

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
10481	2018/10/22 10:09:01	25	2	N/A	Неправильный формат ответа	0	0	Посмотреть	Посмотреть

В этом случае нужно посмотреть протокол проверки, в столбце «Доп. информация» будет написано краткое сообщение проверяющей программы о том, почему решение не было принято на проверку. В данном случае указано, что в решении содержится недопустимый символ «-».

Неправильный формат ответа

N	Результат	Баллы	Доп. информация
1	Неправильный формат ответа	0 (0)	Строка номер 1 содержит недопустимый символ: '-'

Необходимо проверить, какой ответ был сдан в тестирующую систему, для этого нужно вернуться на страницу сдачи задачи и нажать на ссылку «Посмотреть» в столбце «Посмотреть исходный текст».

Исходный код

1	1-3
2	2-7
3	4-6
4	5-7
5	7-8

[Загрузить посылку](#)

После исправления ошибок в записи ответа, нужно отправить на проверку новый ответ по данной задаче.

Сообщение тестирующей системы об ошибках в программе может быть различным. Например, если в строке записаны три числа, а не два, сообщение об ошибке будет следующим.

Неправильный формат ответа

N	Результат	Баллы	Доп. информация
1	Неправильный формат ответа	0 (0)	В строке номер 1 должно быть ровно два числа

В данной задаче в одной строке должны быть два различных числа, если ввести два равных числа в строке, сообщение об ошибке будет следующим.

Неправильный формат ответа

N	Результат	Баллы	Доп. информация
1	Неправильный формат ответа	0 (0)	В строке номер 1 записано два одинаковых числа

Если форма записи ответа удовлетворяет условию задачи, то результат тестирования будет «Принято на проверку». Этот статус не означает, что решение полностью или хотя бы частично верное, решение может быть неправильным, но формат записи ответа был соблюден. Баллы за такие задачи не сообщаются до конца олимпиады.

Участник может сдать несколько ответов по одной задаче, при этом оценивается последнее решение, которое получило статус «Принято на проверку». Если участник сдаёт ещё одно решение, которое было принято на проверку, то у предыдущего принятого на проверку решения статус меняется на «Проигнорирована».

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
10634	2018/10/22 11:45:56	33	2	N/A	Принято на проверку	1	0	Посмотреть	Посмотреть протокол
10625	2018/10/22 11:35:12	25	2	N/A	Проигнорирована	Неизв.	Неизв.	Посмотреть	N/A
10616	2018/10/22 11:29:21	3	2	N/A	Неправильный формат ответа	0	0	Посмотреть	Посмотреть
10615	2018/10/22 11:29:06	3	2	N/A	Неправильный формат ответа	0	0	Посмотреть	Посмотреть
10612	2018/10/22 11:27:36	5	2	N/A	Неправильный формат ответа	0	0	Посмотреть	Посмотреть
10481	2018/10/22 10:09:01	25	2	N/A	Неправильный формат ответа	0	0	Посмотреть	Посмотреть

На изображении выше сначала на проверку были отправлены 4 решения, форма записи ответа которых не соответствовала условию задачи, затем было отправлено решение № 10625, которое было принято на проверку, затем было отправлено ещё одно решение № 10634, которое было принято на проверку, а статус решения № 10634 изменился на «Проигнорирована».

Решения, которые не были приняты на проверку, не будут оценены, для получения баллов по задаче необходимо сдать решение, которое получит статус «Принято на проверку» (при условии, что оно будет правильным).

По ссылке «Итог» отображается информация по всем задачам.

Итог по задачам				
Задача	Статус	Пройдено тестов	Баллы	Номер решения
1	Принято на проверку			10637
2	Принято на проверку			10634
3				
4				
5	Неполное решение	7	6	10646
6				
7				

Суммарный балл: 6

В данном случае на проверку были приняты решения по задачам 1 и 2 с записью ответа. Задачи 5, 6, 7 — задачи по программированию, по этим задачам результат проверки и баллы сообщаются сразу же. Суммарный балл выставлен только по задачам по программированию, по задачам с записью ответа баллы не сообщаются до окончания олимпиады.

Традиционно одна из задач тура является задача на запись формулы. Приведём пример такой задачи.

Из Москвы в Санкт-Петербург выезжает пассажирский поезд, который движется со скоростью v км/ч. Одновременно из Санкт-Петербурга в Москву отправляется скоростной поезд «Сапсан», который движется со скоростью w км/ч. Расстояние от Москвы до Санкт-Петербурга равно s км. Определите, какое расстояние проедет «Сапсан» до встречи с пассажирским поездом. Считайте, что скорости движения поездов постоянные, временем остановок следует пренебречь.

Ответом на эту задачу является некоторое выражение, которое может содержать целые числа, переменные v , w и s (записываемые английскими буквами), операции сложения (обозначается «+»), вычитания (обозначается «-»), умножения (обозначается «*»), деления (обозначается «/») и круглые скобки для изменения порядка действий. Запись вида « $2s$ » для обозначения произведения числа 2 и переменной s неверная, нужно писать « $2 * s$ ».

Пример правильного (по форме записи) выражения: $v + (s - w) * 2$.

Ответ на такие задачи необходимо записать в виде выражения, синтаксис записи которого соответствует общепринятому синтаксису большинства языков программирования. В ответе могут использоваться целые числа, переменные, указанные в условии задачи и арифметические операции «+», «-», «*». Операция деления «/» может быть допустима или нет в зависимости от условия задачи. Порядок действий соответствует общепринятому порядку (сначала — умножения и деления, затем — сложения и вычитания), для изменения порядка действий можно использовать круглые скобки. Пробелы в записи выражения не важны, они могут присутствовать или отсутствовать.

Пример правильного ответа на эту задачу:

$$w * s / (v + w)$$

Правильным является любой ответ, который является эквивалентным данному, форма записи ответа не важна. Например, следующий ответ также будет правильным.

$$s - v * s / (w + v)$$

И даже такой ответ будет верным.

$$s / (w / v + 1) * (w / v)$$

Тестирующая система проверяет ответ, вычисляя значения этого выражения в программе на языке Python, поэтому выражение должно быть корректным. Не нужно добавлять к ответу какие-то слова, знак «=», нельзя использовать скобки другой формы.

Во всех случаях получения результата «Неправильный формат ответа» необходимо изучить протокол проверки и посмотреть на ответ, сданный на проверку.

Типичные ошибки при записи ответа в этой задаче.

Ошибочный ответ	Какая допущена ошибка в записи
$w*s:(v+w)$	Используется «:» вместо «/» для деления
$ц * s / (v + w)$	Вместо буквы «w» набрана русская буква «ц» на той же клавише
$w * s / (v + w).$	В конце записи ответа стоит точка
$w * s / (v + w)$ км	В ответе указана единица измерения «км»
$w * (s / (v + w))$	Некорректное выражение — не закрыта скобка
$w * s / [v + w]$	Используются квадратные скобки, вместо круглых
$w s / (v + w)$	Пропущена операция «*» между переменными w и s
$s = w * s / (v + w)$	В ответе используется символ «=», его не должно быть
Путь равен $w*s/(v+w)$	В ответе используется текст «Путь равен»

Между тем, арифметическое выражение вида «123» с точки зрения тестирующей системы является корректным и будет принято на проверку, но будет оценено в 0 баллов после окончания олимпиады.

После окончания турнира на странице «Итог» отображаются итоговые баллы по всем задачам.

Итог по задачам				
Задача	Статус	Пройдено тестов	Баллы	Номер решения
1	ОК	1	10	10906
2	Неполное решение	0	4	10657
3	Неполное решение	0	0	10638
4				
5	Неполное решение	7	6	10646
6				
7				

Суммарный балл: 20

На странице «Посылки» отображается информация обо всех сданных решениях по всем задачам.

Отправленные решения (последние 15)

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
10906	2018/10/22 13:02:26	15	1	N/A	ОК	1	10	Посмотреть	Посмотреть
10657	2018/10/22 11:59:14	37	2	N/A	Неполное решение	0	4	Посмотреть	Посмотреть
10646	2018/10/22 11:53:27	121	5	python3	Неполное решение	7	6	Посмотреть	Посмотреть
10638	2018/10/22 11:49:14	3	3	N/A	Неполное решение	0	0	Посмотреть	Посмотреть
10637	2018/10/22 11:49:04	3	1	N/A	Проигнорирована	Неизв.	Неизв.	Посмотреть	N/A
10634	2018/10/22 11:45:56	33	2	N/A	Проигнорирована	Неизв.	Неизв.	Посмотреть	N/A
10625	2018/10/22 11:35:12	25	2	N/A	Проигнорирована	Неизв.	Неизв.	Посмотреть	N/A
10616	2018/10/22 11:29:21	3	2	N/A	Неполное решение	0	0	Посмотреть	Посмотреть
10615	2018/10/22 11:29:06	3	2	N/A	Неполное решение	0	0	Посмотреть	Посмотреть
10612	2018/10/22 11:27:36	5	2	N/A	Неполное решение	0	0	Посмотреть	Посмотреть
10481	2018/10/22 10:09:01	25	2	N/A	Неполное решение	0	0	Посмотреть	Посмотреть

[Посмотреть всё](#)

По умолчанию на этой странице отображены последние 15 отправленных решений, чтобы просмотреть все решения, нужно нажать на ссылку «Посмотреть всё».

После окончания олимпиады протокол проверки решения содержит детальную информацию о проверке решения и выставленных баллах. Например, изучим протокол решения № 10657 задачи про упорядочивания списка чисел.

Неправильный ответ

N	Результат	Баллы	Доп. информация
1	Неправильный ответ	4 (10)	[3, 7, 1, 6, 2, 4, 8, 5] - исходное состояние списка [1, 7, 3, 6, 2, 4, 8, 5] - состояни...

==== Тест #1 =====

--- Результат работы: размер 37 ---

```
1 3
2 7
4 6
5 7
7 8
1 2
1 2
```

--- Правильный ответ: размер 0 ---

--- Вывод проверяющей программы: размер 776 ---

```
[3, 7, 1, 6, 2, 4, 8, 5] - исходное состояние списка
[1, 7, 3, 6, 2, 4, 8, 5] - состояние списка после обмена 1
[1, 2, 3, 6, 7, 4, 8, 5] - состояние списка после обмена 2
[1, 2, 3, 4, 7, 6, 8, 5] - состояние списка после обмена 3
[1, 2, 3, 4, 5, 6, 8, 7] - состояние списка после обмена 4
[1, 2, 3, 4, 5, 6, 7, 8] - состояние списка после обмена 5
[2, 1, 3, 4, 5, 6, 7, 8] - состояние списка после обмена 6
[1, 2, 3, 4, 5, 6, 7, 8] - состояние списка после обмена 7
Задача решена за 7 обменов, но есть более короткое решение
```

В разделе «вывод проверяющей программы» приведено состояние списка после каждой операции, которая записана в ответе. В результате был получен правильно упорядоченный список. Но сданное решение содержит 7 операций обмена, в то время как наилучшее решение содержит 5 операций обмена, поэтому это решение оценивается неполным баллом (выставлено 4 балла из 10 возможных).