

Уважаемые участники олимпиады «Успех будущей профессии» в номинации «Программирование»!  
На очном этапе вам будут предложены задачи, решение которых предполагает разработку компьютерных программ. Это пошаговое руководство познакомит вас с задачами и проверяющей системой.

## Вход в систему

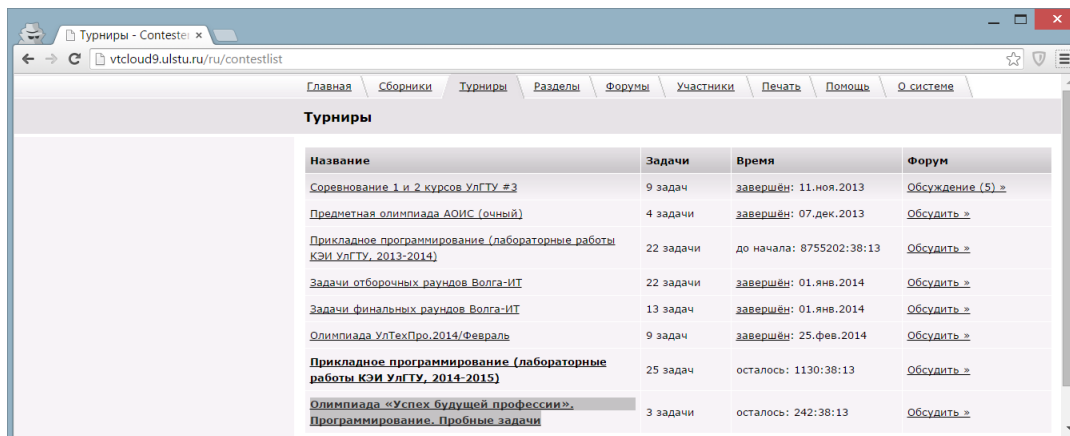
Откройте браузер и перейдите на страницу проверяющей системы: <http://vtcloud9.ulstu.ru/>



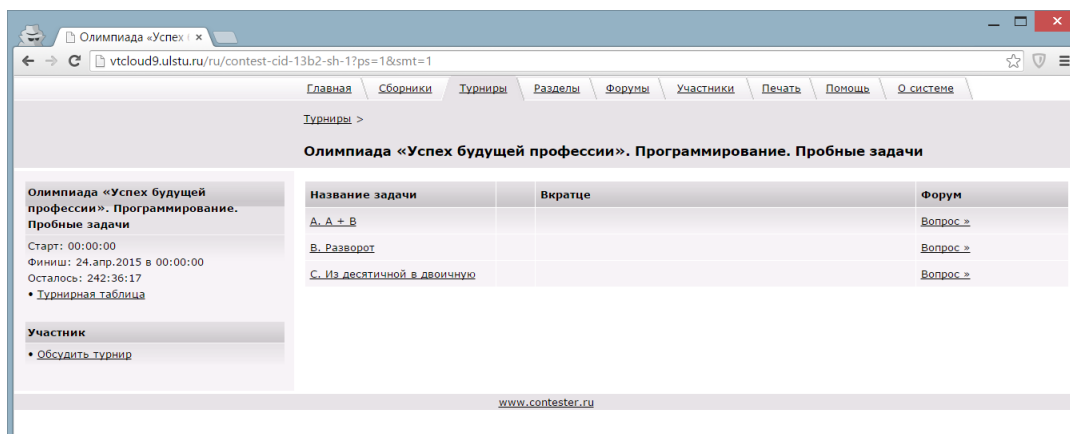
Слева вы видите форму входа. Участники очного тура получат пароли для входа в систему перед началом соревнования. Сейчас выберите одну из учётных записей «test\_0», «test\_1», «test\_2» или «test\_3», введите пароль «00000000» и нажмите кнопку «OK».

## Просмотр задач

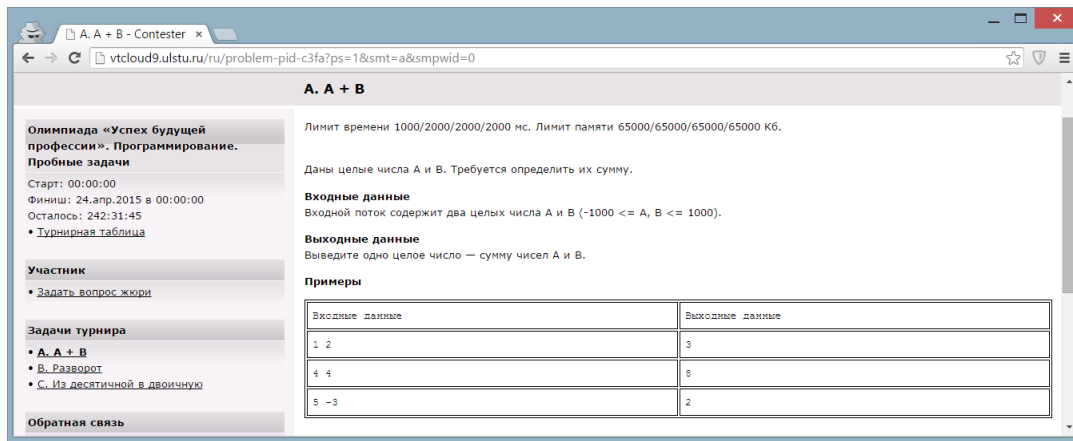
Выберите сверху вкладку «[Турниры](#)».



Откройте турнир «[Олимпиада «Успех будущей профессии». Программирование. Пробные задачи](#)».



Вы увидите список доступных для решения задач. Откройте первую из них – «[A + B](#)».



Вы увидите условие задачи.

## Программирование решений

Для решения каждой из задач вам нужно будет написать программу, которая считывает некоторые входные данные, производит вычисления и выводит соответствующие выходные данные. Например, в этой задаче вам нужно написать программу, которая считывает два числа и выводит их сумму.

Обратите внимание, что в условии задачи чётко указываются ограничения на входные данные (например, здесь написано, что на вход всегда будут поданы ровно два целых числа, каждое из которых не меньше -1000 и не больше 1000), а также даются примеры работы программы (например, если ввести с клавиатуры «1 2», программа должна вывести «3» и закончить работу).

Вы можете писать решения на любом из языков Pascal, C/C++, C# или Java. Вы можете использовать несколько языков, если хотите.

Давайте напишем решение этой задачи на языке Pascal:

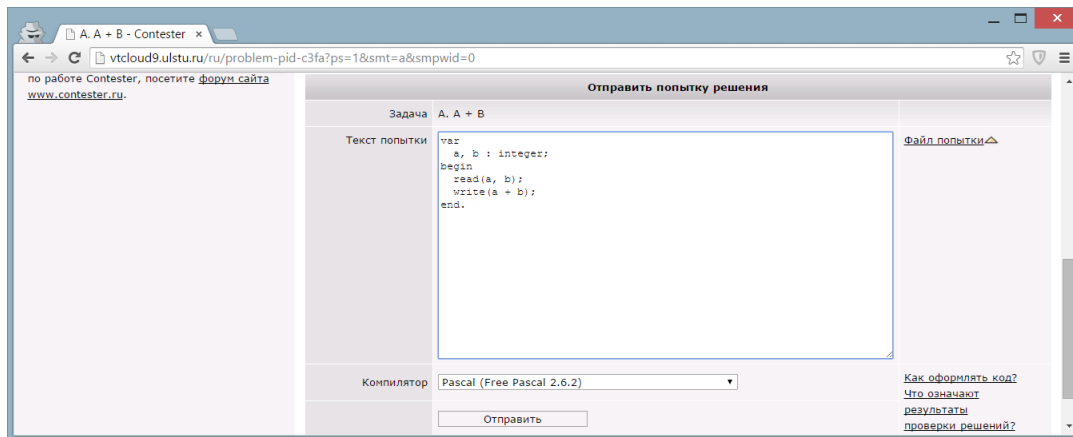
```
var
  a, b : integer; {объявляем переменные под входные данные}
begin
  read(a, b);     {читаем два числа с клавиатуры}
  write(a + b);   {выводим их сумму}
end.
```

Проверим, что программа компилируется без ошибок, запускается и выводит правильные ответы на тесты из условия.

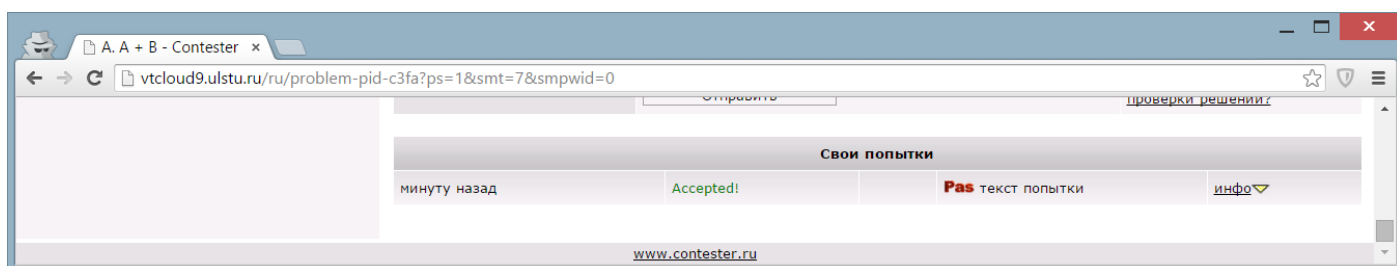
## Проверка решений

Теперь, когда решение задачи готово, мы можем отправить его на проверку. В нижней части окна задачи находится форма для вставки кода решения. Скопируйте туда код, выберите «Pascal» из списка компиляторов и нажмите кнопку «Отправить».

Ваше решение будет проверено сразу же при помощи автоматической системы. Автоматическая система имитирует работу человека: она передаёт программе входные данные, как если бы они вводились с клавиатуры, считывает ответ программы и сравнивает его с правильным ответом.



Кроме тестов, которые показаны в условии, каждая задача содержит ещё несколько (от 10 до 100) секретных тестов, которые не показываются участникам. **Решение задачи считается верным только в том случае, если оно проходит все эти тесты.** Верно решённая задача получает вердикт «Accepted».



## Ошибочные решения

Рассмотрим, что происходит в тех случаях, когда решение содержит ошибки. Поменяйте в коде знак «плюс» на «минус» и отправьте решение заново. Вы получите вердикт «Wrong Answer on test 1» – «Неверный ответ на тесте 1».



Вы сможете увидеть только номер теста, на котором программа вывела неверный ответ, но не содержание этого теста. Тщательно проверяйте свои программы перед отправкой, чтобы не получить вердикт Wrong Answer! Помните, что тесты из условия задачи могут не учитывать какие-либо сложные случаи, на которых ваша программа будет работать неправильно.

Первые тесты во всех задачах совпадают с тестами из условия. Если вы получаете «Wrong Answer on test 1», это означает, что ваша программа на показанном в условии примере выводит другой ответ. В данном случае для теста «1 2» программа вывела ответ «-1», а не «3», так как знак «плюс» мы изменили на «минус».

**Важное замечание.** Не добавляйте в код программы вывод сообщений вида «Vvedite X: » или «Otvet raven: ». Помните, вас проверяет робот, а не человек. Если робот ожидает вывод «123», а ваша программа выведет «Otvet raven: 123», робот засчитает ваш ответ как неправильный.

## Ошибки компиляции

Если в процессе написания кода вы допустите синтаксическую ошибку, и ваша программа не сможет скомпилироваться, вы получите вердикт «Compilation Error» – «Ошибка компиляции». Попробуйте изменить слово «read» на «raed» и отправить код на проверку.



В случае получения вердикта «Compilation Error» вы можете посмотреть отчёт компилятора по ссылке «Инфо – Лог», чтобы понять, какие ошибки были найдены.

## Прочие ошибки

Кроме вердиктов «Accepted», «Wrong Answer» и «Compilation Error», ваше решение может получить следующие:

- «Time Limit» – решение работает слишком долго или «зависло» (например, произошёл вход в бесконечный цикл или программа ожидает лишнее нажатие клавиши);
- «Runtime Error» – решение аварийно завершило работу (деление на ноль, корень из отрицательного числа, обращение по неправильному указателю и т. п.);
- «Presentation Error» – решение вывело слишком много или слишком мало выходных данных, или их формат не соответствует требуемому (например, нужно было вывести число, а выведена строка).

## Подведение итогов

На соревновании решать предложенные задачи можно в любом порядке, с использованием любых доступных языков. При получении сообщения об ошибке можно изменять решение и отправлять его на проверку заново. Количество попыток по любой задаче не ограничено.

Более высокое место в турнирной таблице занимает тот участник, который решил большее число задач. Если два участника решили одинаковое количество задач, выше будет тот из них, кто раньше отправил верные решения и имеет меньшее количество неверных попыток по решённым задачам.

## Получение помощи

Во время соревнования задать вопрос по задаче можно с помощью ссылки «Задать вопрос жюри» в левой части окна задачи.

До начала соревнования различные неясные моменты можно уточнить по e-mail: [v.folunin@gmail.com](mailto:v.folunin@gmail.com)

## Примеры решения задачи «А + В»

Pascal

```
var
  a, b : integer;
begin
  read(a, b);
  write(a + b);
end.
```

C/C++

```
#include <stdio.h>
int main() {
  int a, b;
  scanf("%d%d", &a, &b);
  printf("%d", a + b);
}
```

```
#include <iostream>
using namespace std;
int main() {
  int a, b;
  cin >> a >> b;
  cout << a + b;
}
```

C#

```
using System;
class solver
{
  public static void Main(string[] args)
  {
    string[] tokens = Console.ReadLine().Split(' ');
    int a = Convert.ToInt32(tokens[0]);
    int b = Convert.ToInt32(tokens[1]);
    Console.WriteLine(a + b);
  }
}
```

Java

```
import java.io.*;
import java.util.*;
class solver
{
  public static void main(String[] args) throws IOException
  {
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out));
    StringTokenizer str;
    str = new StringTokenizer(in.readLine());
    int a = Integer.parseInt(str.nextToken());
    int b = Integer.parseInt(str.nextToken());
    out.println(a + b);
    out.flush();
  }
}
```