

# Рекомендации по работе в системе администрирования турниров ejudge

Система ejudge предназначена для проведения турниров и олимпиад по программированию. Её основное назначение – предоставление пользователям удобного интерфейса проведения соревнований с возможностью проверки сдаваемых решений в режиме реального времени. Помимо этого, система может быть использована при проведении учебных занятий. Более подробно об этой системе можно узнать по адресу [ejudge.ru/wiki/index.php/](http://ejudge.ru/wiki/index.php/) Далее приводятся рекомендации для желающих самостоятельно освоить работу в данной системе на практике.

Для участия в пробном туре при помощи тестирующей системы ejudge нужно:

1. Зайти на сайт [olymp.isu.ru](http://olymp.isu.ru).
2. Выбрать пункт “Регистрация на пробный тур”.
3. Выбрать “Create account”
4. Ввести в соответствующие поля свои логин (лучше фамилию и имя латиницей) и актуальный e-mail
5. Нажать “Create account”
6. Появится таблица с указанием логина и сгенерированного для него пароля. Сохранить или записать логин и предоставленный пароль (пароль можно будет потом сменить).
7. Далее, НЕ ВЫХОДЯ С ТЕКУЩЕЙ СТРАНИЦЫ, нажать на ней кнопку “Log in”.
8. ПОДТВЕРДИТЬ РЕГИСТРАЦИЮ: “Confirm registration”.
9. Далее выбрать “Participate” на текущей странице или вернуться на страницу [olymp.isu.ru](http://olymp.isu.ru), выбрать “Участие в пробном туре” и ввести логин-пароль там.
10. Откроется интерфейс участника соревнования “Пробный тур”, задачи которого совпадают с задачами прошлого муниципального этапа по информатике. Это соревнование имеет исключительно ознакомительный характер и не является олимпиадой, выявляющей победителя. Следует выбрать задачу и сдать ее решение. Сдается исходный код программы на языках Pascal, C++, java, python или Basic. Программа должна читать и выводить данные как в стандартный поток так и в указанный файл.
11. Пример решения задачи “Максимум из трех” на языке C++ с использованием стандартного потока:

```
#include <iostream>

using namespace std;

int a, b, c;

int main(){
    cin >> a >> b >> c;
    cout << max(max(a,b),c);
}
```

12. Пример решения задачи “Легкий троллинг” на языке Java с использованием файлового ввода/вывода.

```
import java.io.*;
```

```

import java.util.*;

public class z {
    public static void main(String args[]) throws IOException {
        Scanner fin = new Scanner(new File("input.txt"));
        String s = fin.nextLine();

        PrintWriter fout = new PrintWriter(new File("output.txt"));

        fout.print(s.substring(2,3)+s.substring(0,1)+s.substring(3,4)+s.substring(
1,2));
        fout.close();
    }
}

```

13. Пример решения задачи “Максимум из трех” на языке Pascal с использованием файлового ввода/вывода.

```

var a,b,c: integer;

function max(a,b: integer): integer;
begin
    if a > b
    then
        max:=a
    else
        max:=b;
end;

begin
    assign(input, 'input.txt');
    assign(output, 'output.txt');
    reset(input);
    rewrite(output);
    readln(a, b, c);
    writeln(max(max(a, b), c));
    close(output);

```

end.

14. Пример решения задачи “Максимум из трех” на языке Basic с использованием файлового ввода/вывода.

```
dim a as integer
dim b as integer
dim c as integer
dim m as integer
open "input.txt" for input as #1
open "output.txt" for output as #2
input #1, a, b, c
if a > b then m = a else m = b
if c > m then m = c print #2, trim(str(m))
```

15. Пример решения задачи “Легкий троллинг” на языке Basic с использованием стандартного потока:

```
dim a as String
input a
print trim(chr(a[2]));trim(chr(a[0]));trim(chr(a[3]));trim(chr(a[1]))
```

16. Для того чтобы сдать программу на проверку, нужно в окне задачи из выпадающего списка выбрать язык, на котором написано решение, указать файл с исходным кодом программы и кликнуть “Send”.
17. Обращаем ваше внимание, число попыток сдачи одним участником по одной задаче ограничено десятью попытками.
18. Возможны следующие основные ответы системы:
- “Ошибка компиляции” – отправленная программа имеет синтаксические ошибки, в этом случае в разделе “Посмотреть протокол” можно прочитать отчет об этих ошибках и, исправив их, снова послать решение на проверку. При этом “Ошибка компиляции” не считается полноценной попыткой сдачи и не учитывается в числе попыток.
  - “ОК” или “Неполное решение” – отправленная программа успешно скомпилирована и принята на проверку. Такая попытка уменьшает общее число попыток по данной задаче на одну.

Прежде всего, следует помнить, что регистрация для самостоятельного ознакомления и регистрация для участия в муниципальном этапе по информатике будут различаться. Если в первом случае достаточно проделать приводимые выше шаги, то для участия в МЭ методической комиссией будут сгенерированы пары логин-пароль (с необходимым запасом), которые будут вместе с пакетом заданий разосланы по муниципалитетам. Жюри на местах должны будут распределить эти логины по участникам, зафиксировав распределение документально. Данное распределение одновременно является кодированием участников, так как администраторы системы будут работать с принимаемыми решениями исключительно под полученными участниками логинами. Самостоятельной регистрации школьников для участия непосредственно в МЭ не допускается.

Предлагаемый для ознакомления контекст никоим образом не будет влиять напрямую на результаты муниципального этапа по информатике, в тоже время учащиеся, уверенно

освоившие систему ejudge, в частности, и методику работы с автоматическими проверяющими системами вообще, будут иметь преимущество перед теми, кто не обладает такими умениями, или не тренировал их. Следует особо отметить, что система администрирования турниров ejudge применяется при организации регионального этапа по информатике и ИКТ в Иркутской области, таким образом, участники, желающие добиться высоких результатов, должны будут освоить данную систему на достаточно высоком практическом уровне.

Помимо обучающихся, данную систему рекомендуется освоить и преподавателям информатики, желающим участвовать в подготовке олимпиадников высокого уровня. Для тех учителей, которые будут принимать участие в организации муниципального этапа непосредственно на площадках их проведения, знание основ работы с системой становится обязательным. Все возникающие вопросы, касающиеся работы системы, в случае невозможности их решения самостоятельно, можно направлять по адресу [irk\\_olymp@list.ru](mailto:irk_olymp@list.ru).

## Типичные ошибки участников

Обращаем ваше внимание на следующие типичные ошибки, совершаемые начинающими при работе с автоматическими системами проверки:

### недостаточное внимание к формализму системы

некоторые участники не понимают, что система, как правило, просто сравнивает посимвольно правильный ответ на задачу с ответом, сгенерированным проверяемой программой; если в условии сказано, что числа должны быть разделены строкой, то это так и должно быть; если в условии сказано, что должно быть выведено "Yes" с большой буквы, то "yes" будет признано неправильным ответом, и т.д.

### лишние запросы перед считыванием

многие неопытные участники перед считыванием входных данных, пишут фрагмент кода, подобный следующему:

```
writeln('Введите два числа');  
readln(a,b);  
writeln(a+b);
```

Так как система просто сравнивает то, что проверяемая программа подает на выход, то на выходе она обнаруживает набор символов, начинающийся с "Введите два числа" и засчитывает эту строку как неправильный ответ. Никаких приглашающих к вводу операторов в программе быть не должно. Программа должна начинать свою работу непосредственно со считывания данных в том порядке и формате, в каком это указано в условии.

### излишние проверки границ

для правильного выбора типов данных в условии грамотно оформленной задачи обязаны быть ограничения на все входные параметры (количество вводимых чисел, их тип (натуральные, целые, вещественные с определенной точностью), диапазон из которого их можно брать, длины строк и т.п.). Условие задачи гарантирует, что все входные данные в тестах будут соответствовать этим ограничениям и проверять подходят ли входные данные под эти ограничения в самой программе - нет необходимости. Эта проверка не скажется на результате выполнения программы, но ее написание может отнять лишнее время. С другой стороны следует понимать, что в пределах оговоренных ограничений жюри вправе генерировать тесты любой сложности. И если в условии сказано, что на вход подается не более 100000 целых чисел по модулю не превосходящих  $10^9$ , то, скорее всего, среди тестов встретится и такой, где будут сто тысяч чисел, каждое из которых близко к миллиарду.

### при работе с языком Basic

следует учитывать, что по умолчанию выводится пробел перед числом. Проверяющей системой это может быть в некоторых случаях принято за ошибку. Поэтому рекомендуется при выводе числовых данных использовать команду

```
print trim(str(a))
```

, где  $a$  – выводимое число.

Все эти и подобные им недостатки и недопонимания, допускаемые как учениками, так и их учителями, следует уяснить и устранить при решении задач пробного тура.