

Требования

к организации и проведению муниципального этапа Всероссийской олимпиады школьников по информатике в Вологодской области в 2021/2022 учебном году.

1. Общее положение:

1.1. Возрастные параллели (комплекты заданий).

На муниципальном этапе олимпиады по информатике используется два комплекта заданий – для 7-8 и для 9-11 классов.

1.2. Количество туров и краткое описание (теоретический, практический, творческий, проект, эссе и т.д.).

Муниципальный этап олимпиады по информатике проводится **в один практический тур** с использованием компьютеров.

Для получения логинов и паролей для участников представителю оргкомитета либо жюри каждого муниципального образования необходимо заранее **(желательно не позднее 15.11.2021)** отправить письмо в предметно-методическую комиссию по адресу igand@mail.ru, в котором указать:

- название муниципального образования;
- количество участников от 7, 8, 9, 10 и 11 классов;
- рекомендуется также приложить ФИО участников (поскольку проверка решений выполняется автоматически, необходимости в обезличивании на олимпиаде по информатике нет).

В ответном письме будут отправлены логины и пароли участников, а также ссылка для входа в систему соревнования.

Накануне основного тура рекомендуется провести **пробный тур**, предназначенный для получения навыков работы с тестирующей системой и для проверки участниками своих рабочих мест. Также на пробном туре выдаётся памятка участника. Примерная памятка приведена в Приложении 1.

Перед **началом тура** все участники должны пройти **регистрацию**.

Каждый участник размещается за выделенным ему рабочим местом в соответствии с планом размещения участников, подготовленным оргкомитетом соответствующего этапа.

1.3. Длительность каждого тура (количество минут на выполнение каждого тура, количество минут на выполнение в целом олимпиадных заданий).

Длительность основных туров составляет **3 часа 55 минут для 9-11 классов и 3 часа для 7-8 классов**.

Рекомендуемая длительность пробного тура – 1 час.

1.4. Максимальное количество баллов (за тур, за комплект заданий).

Каждое задание оценивается в 100 первичных баллов независимо от его предполагаемой сложности. Каждый комплект включает 5 заданий. Соответственно, **максимальное количество** первичных баллов за тур равно **500**.

В соответствии с рекомендацией Министерства просвещения РФ **максимум результатов** тура должен быть равен **100 баллам**. Поэтому для определения окончательных баллов участника необходимо разделить сумму его первичных баллов на количество задач. Результат допустимо округлить до двух

знаков после запятой.

1.5. Показ и разбор олимпиадных работ.

Разбор задач.

После тура будет проходить разбор задач. На нём жюри знакомит участников с решениями жюри и/или авторов задач. Авторский разбор задач после проведения олимпиады будет размещён на сайте olympiads.vogu35.ru.

Показ работ и апелляция.

После тура будет проводиться показ работ и апелляция. На показе работ участник, сдававший задачу, но набравший за неё неполный балл, имеет право посмотреть один из не пройденных тестов и ответ на него своей программы.

Претензии к жюри принимаются **только** в том случае, если какой-то тест не соответствует условию задачи или проверяющая система отклонила правильный ответ. Содержание заданий, критерии и методика оценивания не могут быть предметом апелляции и пересмотру не подлежат.

На следующий день после тура участникам станет доступной возможность самостоятельно смотреть все не пройденные тесты своих решений, а также дорешивать задачи вне зачёта. Для этого нужно сохранить карточку с логином, паролем и адресом входа в систему.

1.6. Принцип подведения итогов (по возрастным параллелям или комплектам).

Подведение итогов выполняется отдельно по каждой возрастной параллели.

Списки победителей и призёров утверждаются организатором соответствующего этапа олимпиады.

1.8. Требования к аудиторному фонду.

Аудитории должны соответствовать требованиям, приведённым в разделе 4.

2. Внутреннее содержание заданий по общеобразовательному предмету (принципы формирования комплектов заданий, методические подходы к составлению заданий).

Ввиду того что в начале учебного года небольшое число учащихся 7-8 классов, как правило, владеют навыками программирования, в комплект заданий рекомендуется включать как задания по программированию, так и задания, не требующие навыков программирования. В связи с этим на олимпиаде используются два типа заданий: с кратким ответом (только у 7-8 классов) и задания по программированию с использованием универсальных языков (у всех классов).

2.1. Задания с кратким ответом

Задания в компьютерной форме с кратким ответом представляют собой задания, ответ на которые вводится участником в тестирующую систему и проверяется автоматически. Ответом на такое задание может быть одно или несколько чисел, записанных в одной или нескольких строках, одна или несколько строк текста и т. д. Ответ вводится участником непосредственно в тестирующую систему в поле ввода ответа или записывается в текстовом файле, который сдаётся в тестирующую систему на проверку.

Проверка подобных заданий осуществляется при помощи автоматической

тестирующей системы, поэтому ответ должен быть записан с соблюдением формата записи ответа, указанного в условии задачи. Например, в условии задачи может быть указано, что ответом является ровно пять чисел, записанных через пробел, или последовательность из букв английского алфавита, или последовательность команд исполнителя из фиксированного набора, записанных по одной в строке, или некоторое арифметическое выражение, содержащее числа, переменные, арифметические операции, скобки и т. д.

Примерные темы заданий:

- составление алгоритмов для исполнителя. В условии такой задачи даётся описание исполнителя и его системы команд, ответом на задание является алгоритм для исполнителя. Также ответом могут быть некоторые числа (например, сколько раз исполнитель посетил заданную клетку, и т.п.);
- комбинаторные задачи, например, задачи на составление расписаний, турниров, упорядочивание или подсчёт объектов и т. д. Ответом на такие задачи может быть перестановка объектов, составленное расписание по заданному набору условий, разбиение объектов на несколько групп и т. д.;
- задачи на основы арифметики и теории чисел (делимость, остатки, разложение на простые множители, наибольший общий делитель и т.п.);
- задачи на составление выражений. Ответом на такую задачу является некоторая формула, использующая числа, переменные (описанные в условии задачи), арифметические операции, скобки. Задания такого рода являются введением в программирование, поскольку для их решения необходимо понимание понятий: переменная, операция, порядок вычисления выражения и т. д.;
- логические задачи. Ответом на эту задачу может быть конструкция, удовлетворяющая условиям задачи – например, перечисление, кто из людей является рыцарем, а кто – лжецом, и т. д.;
- задачи на сортировки, взвешивания, перекладывания, переливания, переправы. Ответ на такие задачи можно записать в форме последовательности действий, необходимых для решения задачи, или, например, описать набор гирек, позволяющий выполнить требуемое условие, и т. д.;
- лабиринтные задачи. Ответом на эту задачу может быть последовательность шагов, приводящая к выходу из клетчатого лабиринта. В таких задачах исполнитель при движении по лабиринту может собирать объекты, набирать очки за прохождение через специальные клетки и т. д.;
- выполнение описанного в условии задачи алгоритма;
- кодирование данных. В задачах такого рода необходимо составить код, удовлетворяющий определённым условиям, или закодировать (декодировать) сообщение по описанным правилам;
- обработка файла с данными. В задачах такого рода прилагается файл с данными в текстовом формате, формате CSV или в формате электронных таблиц. Задание заключается в необходимости обработки информации, содержащейся в данном файле, и нахождении ответа на задание. Для выполнения задания можно пользоваться любыми доступными программными средствами (системы программирования, редакторы электронных таблиц, текстовые редакторы и т. д.).

Ответом на задание является одно или несколько чисел, или одна или несколько строк текста.

2.2. Задания по программированию с использованием универсальных языков.

Задачи по программированию должны иметь алгоритмический характер. Задача должна подразумевать ввод данных, обработку их в соответствии с условием, и вывод результата. Формат ввода данных и вывода результата должен быть корректно сформулирован и подробно описан в условии задачи. Рекомендуется использовать наиболее естественные и простые форматы ввода и вывода, чтобы этапы ввода данных и вывода результата не были основной трудностью при решении задачи. Рекомендуется использовать стандартный поток ввода (клавиатура) для ввода данных, стандартный поток вывода (экран) для вывода результатов. Не рекомендуется использовать файловый ввод/вывод. При вводе нескольких чисел или массива рекомендуется вводить каждое число в отдельной строке. Не рекомендуется подавать на вход последовательность данных неизвестной длины, для считывания которой необходимо считывать входной поток до появления признака конца потока.

Условие задачи должно быть сформулировано однозначно, в её формулировке не должно быть неоднозначных трактовок, неполных или противоречивых формулировок.

В тексте условия задачи желательно не использовать термины и понятия, выходящие за пределы школьной программы, при необходимости использования они должны быть определены и конкретизированы.

Если ограничения на входные данные не укладываются в 32-битные знаковые целочисленные переменные, то в условии задачи рекомендуется разместить примечание об этом с указанием того, какие типы данных необходимо использовать для работы с такими переменными в различных языках программирования.

Решением задачи является программа, написанная с использованием одного из предлагаемых на олимпиаде языков программирования.

Желательно, чтобы все задачи школьного и муниципального этапов решались на полный балл на всех основных языках программирования, в том числе на Python.

Методическая комиссия готовит для каждой задачи комплект материалов. Допускается использование задач, ранее использованных на других олимпиадах, но незнакомых школьникам данного региона. Не допускается непосредственное копирование комплектов задач прошлых лет, в том числе комплектов других регионов или муниципалитетов. Материалы задачи должны подразумевать автоматическую проверку с использованием тестирующей системы.

Комплект должен включать:

- условие задачи;
- тесты;
- проверяющую программу;
- основное авторское решение;
- примеры других правильных и неправильных решений;
- разбор задачи.

Условие задачи включает:

- описание задачи;
- формат входных данных;
- формат выходных данных;
- примеры входных и выходных данных;
- ограничение по памяти и пример ограничения по времени;
- информацию о подзадачах и системе оценивания;
- сведения о том, какая информация о результатах проверки решения сообщается участнику

При подготовке материалов задач рекомендуется использовать систему Polygon (polygon.codeforces.com).

Примерные темы заданий:

- Задания на вывод формулы, верной при любых допустимых входных данных.
- Задания на разбор случаев.
- Задания на умение работать с датами и со временем.
- Задания на моделирование описанного в условии задачи процесса.
- Задания на перебор вариантов.
- Задания, требующие обнаружения каких-то закономерностей.
- Задания на анализ строковых данных.
- Задания на обработку числовых массивов и матриц, и др.

В условии каждой задачи должны быть приведены примеры входных и выходных данных для этой задачи. Решение участника запускается на тестах из примеров, приведённых в условии задачи, результат работы на этих тестах сообщается участнику. При наличии технической возможности рекомендуется показывать полный протокол проверки (вывод программы, вывод операционной системы о возникших исключениях, комментарий проверяющей программы в случае неправильного ответа) на тестах из примеров.

Для каждой задачи необходимо предусмотреть возможность получения частичной оценки. Для этого в условии задачи могут быть указаны подзадачи – варианты дополнительных ограничений на входные данные, которые упрощают решение задачи. Альтернативой является потестовая оценка, когда каждый пройденный тест оценивается определённым количеством баллов. Система оценивания каждой задачи указывается в условии задачи.

3. Критерии и методика оценивания выполненных олимпиадных заданий.

Каждая задача оценивается в 100 первичных баллов независимо от её предполагаемой сложности.

Проверка решений выполняется во время тура в автоматическом режиме. Для заданий по программированию выполняется компиляция и запуск решения на наборе тестов, неизвестных участнику. Тест считается пройденным, если программа участника выдала верный ответ, не превысив ограничения по времени работы и памяти. Баллы за задачу начисляются пропорционально количеству пройденных тестов.

Для заданий с коротким ответом проверка правильности также выполняется

автоматически. При этом проверяющая программа может выставить полный или частичный балл согласно критериям, указанным в условии задачи. Например, если участник должен в поле ответа ввести несколько чисел, то полученный балл может определяться пропорционально тому, сколько чисел введено верно.

По окончании проверки решения участнику сообщается набранное количество баллов. В зависимости от задачи могут предоставляться также результаты проверки каждого теста. Для задач с кратким ответом может быть установлено ограничение на количество отправок в систему (чтобы предотвратить подбор ответа), это указывается в условии.

Участник в ходе тура может послать несколько решений одной задачи. Окончательный первичный балл участника за каждую задачу равен максимуму баллов среди всех решений.

4. Материально техническое обеспечение для выполнения заданий МЭ олимпиады.

Каждый участник должен быть обеспечен рабочим местом, оснащённым современным персональным компьютером или ноутбуком. Характеристики компьютеров, предоставленных участникам, должны совпадать либо различаться незначительно. На компьютерах должна быть установлена верная дата и время.

Участникам разрешается использование своих клавиатур и мышей. Клавиатуры и мыши не должны быть программируемыми. Использование клавиатур не должно доставлять дискомфорта другим участникам олимпиады.

Компьютеры должны быть объединены в локальную сеть с доступом к тестирующей системе. Доступ в Интернет рекомендуется запретить, за исключением доступа к серверам тестирующей системы по адресам `contest.yandex.ru` и `codeforces.com`, включая субдомены (`official.contest.yandex.ru`, `contest.codeforces.com` и т.п.).

Рекомендуется также предоставить доступ к сайтам, содержащим справочную информацию по основным языкам программирования (C++, Python, Java, Pascal), либо разместить справочные материалы на компьютерах участников или в общей сетевой папке:

- документация по языку C++: <http://cppreference.com>
- документация по языку Free Pascal: <https://www.freepascal.org/docs.var>
- документация по языку Java: <https://docs.oracle.com/en/java/>
- документация по языку Python: <https://docs.python.org/3/>

Следует проверить, что из браузеров на компьютерах участников получается зайти на сайты `contest.yandex.ru` и `codeforces.com`. При необходимости нужно обновить сертификаты безопасности операционной системы либо установить браузер с собственным хранилищем сертификатов (например, Mozilla Firefox).

На компьютере каждого участника необходимо установить программное обеспечение из таблицы 1 с учётом используемой на рабочих местах операционной системы (например, при использовании Linux необходимо установить лишь те языки и среды, которые имеют версию для этой ОС). Допускается использование более поздних версий ПО по сравнению с указанными в таблице.

Допускается заранее собрать заявки от участников о том, какое программное обеспечение нужно каждому из них, и установить именно его.

Рекомендуется дополнительно установить на рабочих местах двухпанельный файловый менеджер (Far, Total Commander) и программу для чтения pdf-файлов.

Таблица 1. Языки и среды программирования

Язык	Транслятор	Среда программирования
C/C++	GNU C/C++ 17 7.3	CodeBlocks 20.03
Object Pascal	Free Pascal 3.0.0	Встроенная, Lazarus 1.6.0
C#	Microsoft Visual C# 2017, Mono C# 5.2.0	Встроенная
C/C++	Microsoft Visual C++ 2017	Встроенная
Python 3	Python 3.7.2	IDLE или Wing IDE 101, PyCharm Community Edition
Java	Oracle Java 8	Eclipse JDT , IntelliJ IDEA Community Edition
Pascal	Pascal ABC.NET 3.5.1	Встроенная

Задания тиражируются на листах бумаги формата А4 или А5. Допускается предоставление доступа к электронным версиям заданий в интерфейсе тестирующей системы.

Учащимся предоставляется бумага и письменные принадлежности для черновых записей. При этом черновики не собираются после окончания тура и не проверяются.

5. Примеры олимпиадных заданий, ответов и решений.

Пример задачи 1. Числовая последовательность.

Крокодил Гена очень любит математику. Особенно ему нравятся числовые последовательности, он с удовольствием их придумывает и изучает. К своим занятиям он привлекает и Чебурашку. Однажды Гена выписал вот такую последовательность: 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5... После этого он попросил Чебурашку ответить на вопрос: на каком месте в этой последовательности впервые встретится число N (члены последовательности нумеруются с единицы)? Вам нужно найти формулу, отвечающую на данный вопрос.

Формат ответа. Напишите формулу в поле ответа. В формуле разрешено использовать только следующие элементы:

- переменная N
- целые числа в диапазоне от -1000 до 1000
- круглые скобки
- знаки операций +, -, *, /, где '+' означает сложение, '-' – вычитание, '*' – умножение, '/' – деление нацело (например, $5/3 = 1$).

Примечание: приоритет операций умножения и деления нацело выше, чем у операций сложения и вычитания.

Пример ответа (этот ответ неправильный):

$$(N + 5) / (N - 1) * 2$$

Система оценивания. Правильность вашей формулы будет проверяться подстановкой в неё нескольких наборов значений переменных из следующего

диапазона: $1 \leq N \leq 100$. За каждый верный ответ начисляется определённое количество баллов.

Решение. Несложно заметить, что искомый номер можно найти как сумму арифметической прогрессии, и ответом будет формула $N * (N + 1) / 2$

Важно то, что операция деления стоит именно в конце, так как деление целочисленное. Формулы вида $N / 2 * (N + 1)$ или $(N + 1) / 2 * N$ будут давать неверные ответы при нечётном N или нечётном $(N+1)$ соответственно.

Пример задачи 2. Игра.

Однажды Баба-Яга пригласила в гости Змея Горыныча и предложила ему сыграть в очень простую игру. Она взяла чётное количество карточек, на каждой из которых написано произвольное натуральное число, не превышающее 1000, тщательно перемешала все карточки и разложила их на столе числами вниз.

Первым ходом Баба-Яга открывает ровно половину карточек, вторая половина карточек достаётся Змею Горынычу. Затем каждый игрок считает сумму чисел, написанных на его карточках. Выигрывает тот, у кого полученная сумма оказывается больше. Выигрыш определяется разностью между большим и меньшим из полученных чисел.

Однако, хитрая Баба-Яга не предупредила Змея Горыныча о том, что у нее имеются волшебные очки, которые позволяют видеть надписи сквозь бумагу любой плотности, и она не забывает их надевать прежде, чем приступить к игре. Напишите программу для определения максимальной суммы выигрыша, которую может получить Баба-Яга, с учетом того, что ей видны все числа, написанные на карточках.

Входные данные. Первая строка содержит одно чётное натуральное число N – количество карточек. Вторая строка содержит ровно N натуральных чисел, разделённых одним пробелом.

Выходные данные. Выведите одно целое число – максимальный выигрыш, который может получить Баба-Яга.

Пример ввода

4

2 4 7 1

Пример вывода

8

Система оценивания.

Подзадача 1 (до 50 баллов): $2 \leq N \leq 100$

Подзадача 2 (до 50 баллов): $100 < N \leq 10^5$

Решение. Так как надо найти наибольший выигрыш, значит, хитрая Баба-Яга должна взять карточки с самыми большими числами, а Змею Горынычу достанутся все остальные. Для решения этой задачи можно воспользоваться одним из алгоритмов сортировки. После того как числа в массиве станут упорядочены по убыванию, достаточно из суммы чисел, расположенных в первой половине массива, вычесть сумму чисел, расположенных во второй половине.

Чтобы решить эту задачу на полный балл, необходимо использовать эффективный алгоритм сортировки, работающий быстрее, чем за квадрат.

Простейший вариант - использовать готовую функцию сортировки из стандартной библиотеки языка программирования.

Пример решения на C++:

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

int main() {
    int n;
    scanf("%d", &n);
    std::vector<int> a(n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }
    std::sort(a.begin(), a.end());
    int sum1 = 0, sum2 = 0;
    for (int i = 0; i < n / 2; i++)
        sum1 += a[i];
    for (int i = n / 2; i < n; i++)
        sum2 += a[i];
    printf("%d\n", sum2 - sum1);
}
```

Другие примеры олимпиадных задач прошлых лет, ответов и решений можно найти на сайте olympiads.vogu35.ru