Тамбовское областное государственное образовательное автономное учреждение дополнительного профессионального образования «Институт повышения квалификации работников образования»

ТРЕБОВАНИЯ К ПРОВЕДЕНИЮ МУНИЦИПАЛЬНОГО ЭТАПА ПО ИНФОРМАТИКЕ ВСЕРОССИЙСКОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ В 2019/2020 УЧЕБНОМ ГОДУ

Тамбов 2019

Оглавление

1. Общие положения

Настоящие методические рекомендации подготовлены центральной предметно-методической комиссией (ЦПМК) по информатике и являются частью нормативно-правового обеспечения всероссийской олимпиады школьников. Муниципальные предметно- методические комиссии разрабатывают требования к проведению школьного этапа олимпиады, региональные предметно-методические комиссии разрабатывают требования к проведению муниципального этапа, руководствуясь

- Порядком проведения всероссийской олимпиады школьников
- Региональным порядком проведения этапов всероссийской олимпиады школьников.
- Настоящими методическими рекомендациями.

Муниципальные и региональные предметно-методические комиссии разрабатывают задания для проведения школьного и муниципального этапов всероссийской олимпиады школьников, используя настоящие методические рекомендации.

Для учащихся для учащихся 7-8 классов проводятся школьный и муниципальный этап, для учащихся 9-11 классов проводятся школьный, муниципальный, региональный и заключительный этапы олимпиады.

Учащиеся 5-8 классов вправе выполнять задания за более старшие классы, в этом случае они могут принять участие во всех этапах олимпиады, которые проводятся для соответствующих классов. При этом участие за более старший класс должно начинаться со школьного этапа, поэтому

- учащимся 5-8 классов, которые на уроках, на дополнительных занятиях в кружках или учреждениях дополнительного образования, либо по итогам самообразования продемонстрировали высокий уровень программирования на универсальных языках общего назначения (C++, Python, Pascal, Java, C#) и проявляют интерес к решению алгоритмических задач по программированию (например, систематически участвующие в соревнованиях на codeforces.com или аналогичных сайтах, решающие задачи на сайтах с архивами задач вида informatics.msk.ru, acmp.ru, acm.timus.ru, и др., принимавшие участие в летних школах или сборах по решению задач по программированию), рекомендуется принимать участие в олимпиаде за 9 класс, начиная со школьного этапа, с возможностью участия в региональном и заключительном этапе;
- учащимся 5-6 классов, проявляющим интерес к информатике, дополнительно занимающимся информатикой в кружках, учреждениях дополнительного образования или в форме самообразования, знакомым с формой проведения и уровнем заданий муниципального этапа за 7 класс в данном регионе, рекомендуется принимать участие в олимпиаде за 7 класс с возможностью участия в муниципальном этапе;
- учащимся, знакомство которых с информатикой ограничивается школьными уроками, рекомендуется принимать участие в школьном этапе за свой класс обучения.

2. Принципы формирования комплектов олимпиадных заданий

2.1. Школьный и муниципальный этапы для 7-8 классов

Для учащихся 7-8 классов проводятся школьный и муниципальный этапы олимпиады. Рекомендуется проведение олимпиады в один тур, продолжительность тура школьного и муниципального этапов составляет от 90 до 180 минут.

Школьный и муниципальный этапы олимпиады рекомендуется проводить с использованием автоматической тестирующей системы для ввода и проверки решений участников, например Яндекс-контест contest.yandex.ru, Ejudge ejudge.ru, и др. Для проведения олимпиады рекомендуется использовать задания нескольких видов из числа следующих:

- Компьютерная форма заданий с кратким ответом - задания, ответ на которые записывается в виде одного или нескольких чисел, одной или нескольких строк текста.
- Задания на использование компьютерных сред для формальных исполнителей или виртуальных лабораторий.
- Задания по программированию с использованием универсальных языков, таких как Pascal, Python, C++, Java, C# и т.д.

Ввиду того, что в начале учебного года небольшое число учащихся 7-8 классов, как правило, владеют навыками программирования, в комплект заданий рекомендуется обязательно включать задания как по программированию, так и задания, не требующие навыков программирования, то есть задания олимпиады должны быть доступны и интересны учащимся с различным уровнем подготовки по информатике и программированию, в том числе только начинающим изучать информатику.

Задания, требующие навыков использования какой-либо конкретной учебной среды программирования (например, Scratch или Логомиры) могут предлагаться по решению муниципальной или региональной предметно-методической комиссии только если во всех образовательных учреждениях данного муниципального образования или региона созданы условия для изучения данной среды, то есть такие задания должны быть доступны всем обучающимся.

Рекомендуется включать в вариант школьного и муниципального этапов 4-6 заданий различной тематики и различного уровня сложности. Первая задача должна быть доступна практически всем участникам олимпиады, далее сложность заданий должна возрастать. Сложность последней задачи должна быть такой, чтобы её решали участники уровня победителя соответствующего этапа олимпиады.

Возможно составление варианта из большего числа заданий, если вариант составляется из заданий различной формы (например, задания как по программированию, так и задания с вводом ответа), чтобы дать возможность учащимся с различным уровнем подготовки в области программирования проявить свои способности. В этом случае окончательный балл можно выставлять не по сумме баллов за все задачи, а по сумме баллов за фиксированное число задач, по которым получен наилучший результат.

2.2. Школьный и муниципальный этапы для 9-11 классов

Для учащихся 9-11 классов проводятся школьный и муниципальный этапы олимпиады. Далее участники муниципального тура, набравшие необходимое для участия в региональном этапе олимпиады количество баллов, установленное организатором регионального этапа олимпиады, принимают участие в региональном этапе олимпиады. С учетом этого рекомендуется проведение олимпиады в формате, приближенном к региональному этапу, но с учетом более широкого охвата участников.

Рекомендуется проведение олимпиады в один тур, продолжительность тура школьного и муниципального этапов составляет от 120 до 240 минут.

Школьный и муниципальный этапы олимпиады рекомендуется проводить с использованием автоматической тестирующей системы, как правило, той же, что будет использоваться на региональном этапе в данном регионе.

Для проведения олимпиады рекомендуется использовать задания по программированию с использованием универсальных языков, таких как Pascal, Python, C++, Java, C# и т. д.

Рекомендуется включать в вариант школьного и муниципального этапов 4-6 заданий различной тематики и различного уровня сложности. Первая задача должна быть доступна практически всем участникам олимпиады, далее сложность заданий должна возрастать. Сложность последней задачи должна быть такой, чтобы её решали участники уровня победителя соответствующего этапа олимпиады.

При составлении варианта не рекомендуется включать задачи, требующие знания специфических алгоритмов, например алгоритмов на графах, алгоритмов на строках, алгоритмов динамического программирования. В любом случае не следует включать более 1-2 таких задач, они должны быть максимальными по сложности, помимо таких задач в комплект должны входить не менее 4 задач, не требующих знания специфических алгоритмов.

С другой стороны, не рекомендуется ограничиваться только задачами, единственной трудностью которых является реализация описанных в условии задачи действий, или задачами, решение которых полностью заключается в выводе математической формулы. Такие задачи могут входить в комплект, но необходимо также включать в комплект задачи, решение которых сочетает математическую или алгоритмическую идею и реализацию вычислений, необходимых для получения ответа, с использованием возможностей выбранного языка программирования.

3. Задания олимпиады

3.1. Материально-техническое обеспечение при компьютерной форме проведения этапа

Каждый участник должен быть обеспечен рабочим местом, оснащенным современным персональным компьютером или ноутбуком. Характеристики компьютеров, предоставленных участникам, должны совпадать, либо различаться незначительно. Компьютеры должны быть объединены в локальную сеть с доступом к

тестирующей системе.

Предметно-методическая комиссия может принять решение разрешить участникам использование своих клавиатур и мышей. Клавиатуры и мыши не должны быть программируемыми. Использование клавиатур не должно доставлять дискомфорт другим участникам олимпиады. На используемые клавиатуры и мыши могут быть наложены дополнительные требования.

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

3.2. Задания в компьютерной форме с кратким ответом

3.2.1. Принципы составления заданий

Задания в компьютерной форме с кратким ответом представляют собой задание, ответ на которое вводится участником в тестирующую систему и впоследствии 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. Между тем, само задание не требует компьютера для выполнения.

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.

3.2.2. Тематика заданий

Примерные темы заданий:

- 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.
- Комбинаторные задачи, например, задачи на составление расписаний, турниров, упорядочивание или подсчет 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 сортировки, взвешивания, перекладывания, переливания, переправы. Ответ на такие задачи можно записать в форме последовательности действий, необходимых для решения задачи, или 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 прохождения через специальные клетки и т. д.
- Составление алгоритмов для исполнителя. В условии такой задачи даётся описание исполнителя и его системы команд, ответом на задание является алгоритм для исполнителя.
- 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.

3.2.3. Материально-техническое обеспечение

На компьютерах должна быть установлена программа для доступа в тестирующую систему (например, браузер, если доступ к тестирующей системе осуществляется через web- интерфейс).

Задания тиражируются на листах бумаги формата А4 или А5, возможно также предоставлять условия задач только в электронном виде в тестирующей системе. Для черновых записей участникам предоставляется бумага, черновики не сдаются и не проверяются.

3.2.4. Критерии и методики оценивания

Для проверки решений используется автоматическая тестирующая система. Для проверки решения каждой задачи необходимо реализовать проверяющую программу, которая выдаёт для решения один из следующих статусов:

- ―Неправильный формат записи ответа‖.
- ―Полное или частичное решение‖. В этом случае проверяющая программа также возвращает балл, которым оценивается данное решение (от 0 до максимально возможного балла за задачу).
- Возможны и другие варианты статусов, например, ―Неверное решение‖, ―Полное решение‖, ―Частичное решение‖. Все задачи оцениваются одинаковым числом баллов.

При сдаче решения в тестирующую систему производится проверка на соблюдение формата записи ответа, если проверка не пройдена, решение не принимается на проверку и в тестирующей системе указывается статус ―Неверный

формат записи ответа‖. В этом случае желательна выдача дополнительного комментария тестирующей системы о несоответствии сданного ответа формату, описанного в условии задачи.

Окончательная проверка решений с выставлением баллов может производиться как сразу же после сдачи заданий (онлайн-проверка), так и после окончания тура (оффлайн-проверка). Порядок проведения проверки должен быть доведён до сведения участников до начала олимпиады. Следует учесть, что в случае онлайн-проверки возможен подбор ответа участниками олимпиады путём многократной отправки различных решений, поэтому онлайн- проверка возможна только для некоторых видов задач.

Задачи должны предусматривать возможность выставления частичных баллов за сданное решение, однако при автоматической проверке невозможно оценить корректность рассуждения и доказательства, поэтому формулировка задачи должна указывать на возможность выставления частичных баллов. Например, в формулировке условия задачи могут присутствовать фразы ‒Чем меньше команд будет содержать алгоритм, тем больше баллов вы получите‖ или ―Чем меньше гирек будет в предложенном наборе, тем больше баллов вы получите‖ и т. д.

Рассмотрим несколько подходов к методике выставления частичных баллов за такие задачи.

Если ответом на задачу является формула, то проверяющая программа должна принимать любую формулу, эквивалентную правильному ответу. Для этого можно вычислять значение формулы-ответа участника на разных значениях переменных и сравнивать со значением формулы правильного ответа. Неполный балл можно выставлять за формулы, дающие правильный ответ только в частных случаях или при типичных ошибках в составлении формулы, например, при ошибках в формулах на ±1.

Если ответом является некоторая конструкция (перестановка, код, расписание турнира) и т.д., при этом в условии сказано, что оценивается эффективность найденного решения по некоторому параметру (суммарная длина кодовых слов, количество туров в расписании турнира, количество выполненных условий для найденной перестановки и т.д.), то полный балл выставляется за наилучшее возможное решение, частичные баллы выставляются за верное, но не наилучшее решение. Проверяющая программа проверяет ответ на корректность, в случае, если ответ корректен, оценивается его эффективность в соответствии с условием задачи.

Если ответом является алгоритм для исполнителя, маршрут в лабиринте и т.д., баллы могут начисляться в зависимости от количества команд в алгоритме, длине найденного маршрута, количеству очков за пройденные специальные клетки и т.д. Проверяющая программа устанавливает корректность алгоритма или маршрута, в случае его корректности баллы выставляются в зависимости от эффективности решения или числа набранных очков.

Задача может состоять из нескольких независимых заданий с общим условием. Например, дана строка из символов I, V, X, L, C, D, M, нужно разбить её на части, являющиеся корректными римскими числами с минимальной суммой. В такой задаче можно предложить несколько независимых примеров заданий разной сложности, например, первый пример состоит из символов I-X, второй пример из I-C, третий пример из I-M. Каждый пример оценивается независимо, оценка за задание

складывается из суммы баллов за каждый пример.

3.3.Задания на использование компьютерных сред для формальных исполнителей или виртуальных лабораторий

3.3.1. Принципы составления заданий

Задания такого рода выполняются непосредственно на компьютере с использованием среды для составления алгоритма для исполнителя или виртуальной лаборатории для моделирования каких-либо процессов (переливания, взвешивания, управление транспортом и т.д.). В задании требуется составить алгоритм для исполнителя (например, выйти из лабиринта, собрать все объекты в лабиринте, расставить объекты по нужным местам, отмерить нужное число воды, определить массу груза и т.д.).

3.3.2. Тематика заданий

Примерные варианты лабораторий и исполнителей:

- Сортировка объектов
- Взвешивания
- Перемещение объектов (например, движение транспорта)
- Переливания
- Исполнитель ―Робот‖ и его вариации (Лайтбот, Сокобан).
- Исполнитель ―Черепашка‖

3.3.3. Материально-техническое обеспечение

Каждому участнику предоставляется персональный компьютер с установленной на него средой для выполнения заданий.

Среда для выполнения задания может быть интегрирована с тестирующей системой, используемой для сдачи и проверки решений, например, задания могут исполняться непосредственно в браузере, или же быть отдельной программой. В этом случае среда для выполнения задания должна сохранять ответ участника в виде текста или файла, который потом сдаётся в тестирующую систему для проверки.

3.3.4. Критерии и методики оценивания

Задание должно предусматривать возможность выставления частичного балла в зависимости от эффективности решения (количество команд в алгоритме, количество выполненных операций, длина маршрута, пройденного исполнителем, количество собранных на маршруте очков и т.д.).

Проверку подобных заданий желательно производить автоматически при помощи тестирующей системы, проверяющая программа устанавливает корректность сданного решения и оценивает его эффективность на основании критериев, составленных предметно- методической комиссией.

При отсутствии технической возможности для автоматической проверки, решения могут проверяться членами жюри.

3.4. Задания по программированию для решения с использованием универсальных языков

3.4.1. Формирование списка языков программирования

Предметно-методическая комиссия (муниципальная для школьного этапа, региональная для муниципального этапа) формирует список языков программирования, доступных для решения задач. В список рекомендуется включить распространенные языки программирования общего назначения, в том числе:

- C++;
- Pascal;
- Python;
- Java;
- C#.

Не рекомендуется ограничивать участников небольшим количеством доступных языков программирования, в частности, в список могут быть добавлены языки, поддерживаемые используемой тестирующей системой, которые используются для преподавания в школах муниципалитета или региона, например, Basic, КуМир, Kotlin, C, D, и другие.

3.4.2. Принципы составления заданий

Задачи должны иметь алгоритмический характер.

Задача должна подразумевать ввод данных, обработку их в соответствии с условием задачи, и вывод результата. Формат ввода данных и вывода результата должны быть корректно сформулированы и подробно описаны 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 ее 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.

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

Методическая комиссия готовит для каждой задачи комплект материалов. Допускается использование задач, ранее использованных на других олимпиадах, но не знакомых школьникам данного региона. 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 может, например, использоваться система Polygon polygon.codeforces.com, дополнительные методические рекомендации по разработке задач приведены в приложении 2.

3.4.3. Тематика 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
- Задание требующие обнаружения каких-то закономерностей
- Задания на анализ строковых данных
- Задания на обработку числовых массивов

3.4.4. Методика проверки заданий

Решением задачи является программа, написанная на одном из доступных на олимпиаде языков программирования. Для проверки и оценивания решений жюри использует автоматическую тестирующую систему.

На проверку отправляется исходный текст программы. При отправке решения на проверку участник указывает, с использованием какого языка программирования и компилятора выполнено решение. Разные решения, отправленные на проверку, могут использовать разные языки программирования и/или компиляторы.

Присланная программа компилируется с использованием строки компиляции, установленной жюри. Если компиляция завершается неудачно, участнику сообщается, что результат проверки его решения – Compilation Error.

Программа запускается на тестах. Для каждого теста, на котором был выполнен запуск, устанавливается результат выполнения на этом тесте. Верный ответ на тест, выданный при соблюдении указанных в условии задачи ограничений, соответствует результату OK. Для неверных ответов возможны различные результаты выполнения, в зависимости от ошибки, например:

- Wrong answer – неверный ответ на тесте;
- Runtime error – ошибка выполнения на тесте, либо ненулевой код возврата;
- Time limit exceeded – превышено ограничение времени на тесте;
- Memory limit exceeded – превышено ограничение по памяти на тесте.
- Допускаются другие варианты результата проверки на тесте.

Когда программа запускается, ей указанным в условии задачи способом передаются входные данные. Наиболее типичным является использование для ввода данных стандартного потока ввода или текстового файла с определенным в условии задачи именем, размещенного в каталоге запуска.

Сделанный программой описанным в условии задачи способом вывод сохраняется и проверяется с использованием разработанной предметно-методической комиссией проверяющей программы.

При запуске программы участника тестирующая система контролирует время работы решения и использованную память.

В условии каждой задачи должны быть приведены примеры входных и выходных данных для этой задачи. Решение участника запускается на тестах из примеров, приведенных 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.

3.4.5. Методика оценивания заданий

Каждое задание оценивается из максимального балла, указанного в условии задачи или в других документах, доступных участникам - листа с информацией о задачах, правил олимпиады, памятки участника, и т. п. Рекомендуется оценивать все задачи из одинакового максимального балла, например 100 баллов.

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 определенным количеством баллов.

Система оценивания каждой задачи указывается в условии задачи. Если используются общие схемы оценивания в разных задачах, например, подзадачи и зависимости между ними, информация об этом может быть указана в других документах, доступных участникам - листе с информацией о задачах, правилах

олимпиады, памятке участника и т. п.

При использовании потестовой оценки каждый тест оценивается отдельно указанным в условии задачи числом баллов. Балл участника за задачу равен сумме баллов за тесты. В условии задачи могут быть указаны характеристики набора тестов, например, доля или суммарный балл тестов, подходящих под некоторые дополнительные ограничения.

При использовании подзадач тесты к задаче разбиваются на группы, каждая группа соответствует одной подзадаче. Для каждой подзадачи устанавливается её стоимость в баллах. Участник получает баллы за подзадачу, если все тесты группы для этой подзадачи пройдены. В условии задачи могут быть указаны дополнительные ограничения на начисление баллов за подзадачу, например, требование прохождения тестов необходимых подзадач.

Допускается комбинированная система оценивания, когда за некоторые подзадачи баллы начисляются только в случае прохождения всех тестов, а в других подзадачах используется потестовая оценка. Информация об этом должна быть указана в условии задачи.

Для школьного этапа в качестве основной рекомендуется потестовая система оценки. Исключения составляют задачи с ответами вида ―Да\нет‖ и т.п.

3.4.6. Использование тестирующей системы

Жюри школьного или муниципального этапа может установить и настроить собственный экземпляр тестирующей системы, либо использовать тестирующую систему, доступную по модели ―software as a service‖, например:

- Яндекс-контест contest.yandex.ru
- Codeforces codeforces.com

Поскольку администрирование тестирующей системы, даже при отсутствии необходимости локальной установки и настройки, может представлять трудности для жюри школьного или муниципального этапа, рекомендуется централизованная организация тестирования решений на уровне региона.

При проведении муниципального этапа региональная предметно-методическая комиссия может предложить помощь в организации тестирования, поскольку задания одинаковые для всех муниципалитетов.

При проведении школьного этапа задания разрабатываются муниципальными предметно-методическими комиссиями. Для организации централизованного тестирования могут быть организованы отдельные соревнования для каждого муниципалитета. Также, если муниципальные предметно-методические комиссии приходят к соглашению о разработке единых заданий для школьного этапа в регионе, может быть проведено единое соревнование по аналогии с муниципальным этапом.

3.4.7. Необходимое материально-техническое обеспечение

В дополнение к материально-техническому обеспечению, указанному в разделе 3.2 на компьютерах участников должны быть установлены компиляторы и среды разработки для используемых на соответствующем этапе языков программирования.

Ссылки на ресурсы в интернете, содержащие компиляторы и среды разработки, указаны в приложении 3

Помимо ОС, компиляторов и сред разработки на компьютерах участников может быть установлено дополнительное ПО (файловые менеджеры, текстовые редакторы, программы для чтения PDF-файлов), например:

- Far Manager;
- Vim;
- Sublime Text;
- Geany;
- Adobe reader.

4. Перечень справочных материалов, средств связи и электронно-вычислительной техники, разрешенных к использованию во время проведения олимпиады

Помимо компьютера, предоставленного организаторами соответствующего этапа в случае его проведения в компьютерной форме, участникам запрещается пользоваться любыми электронными устройствами, в том числе другими компьютерами и ноутбуками, мобильными телефонами и смартфонами, электронными книгами, планшетами, электронными часами, CD и MP3 плеерами, любыми наушниками.

Участникам запрещается пользоваться любыми электронными носителями информации, в том числе компакт-дисками, модулями флэш-памяти, картами памяти.

Участникам разрешается пользоваться чистыми листами, в том числе листами в клетку, а также письменными принадлежностями – ручкой, карандашом, стирательной резинкой, циркулем, линейкой.

Для каждого основного языка программирования или среды виртуальных исполнителей на компьютерах участников или в локальной сети размещается документация. Также рекомендуется установить или сделать доступной документацию по дополнительным языкам программирования. Допустимо также при ограничении доступа в интернет сохранить доступ к сайтам с документацией по языкам программирования.

5. Порядок проведения туров

5.1. Процедура регистрации участников олимпиады

Перед началом тура все участники должны пройти регистрацию.

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

5.2. Правила поведения участников во время тура

В случае использования компьютеров для проведения этапа перед началом каждого тура все компьютеры участников должны находиться во включенном

состоянии.

На каждом рабочем месте участника должны размещаться распечатанные тексты условий задач (если они используются, допускается использование электронной версии условий, в этом случае они должны быть доступны в интерфейсе проверяющей системы) и лист с логином и паролем для входа в тестирующую систему (если для авторизации используются логин и пароль). В распоряжение участников также должна предоставляться памятка участника. Возможно также предоставление указанных материалов в электронном виде.

Участникам разрешается ознакомиться с условиями задач и приступить к их решению только после начала тура. Распечатанные тексты условий задач должны быть размещены таким образом, чтобы участники не могли свободно ознакомиться с ними до начала тура, например, упакованы в непрозрачный конверт или размещены лицевой стороной вниз.

Во время тура участники не вправе общаться друг с другом или свободно перемещаться по аудитории. Выход из места проведения олимпиады и вход в него во время тура возможен только в сопровождении дежурного.

Участникам категорически запрещается перед началом и во время туров передавать свои логин и пароль другим участникам, пытаться получить доступ к информации на компьютерах других участников или пытаться войти в тестирующую систему от имени другого участника.

В случае возникновения во время тура сбоев в работе компьютера или используемого программного обеспечения время, затраченное на восстановление работоспособности компьютера, может быть компенсировано по решению жюри, если сбой произошел не по вине участника.

Ответственность за сохранность своих данных во время тура каждый участник несет самостоятельно. Чтобы минимизировать возможные потери данных, участники должны своевременно сохранять свои файлы.

5.3. Показ олимпиадных работ

В случае использования онлайн-тестирования, при котором результаты проверки решений сообщаются участникам во время тура по мере того как они становятся известны, участники после окончания тура знают свои результаты.

Организатор соответствующего этапа публикует на своём сайте задания олимпиады и разбор задач. В случае компьютерного проведения тура также публикуются тесты и решения, подготовленные предметно-методической комиссией, *возможно* предоставление *возможности* решения задач вне зачета после окончания тура.

В случае бланковой формы проведения тура участники могут ознакомиться с результатами проверки своих работ.

Не допускается изменение баллов участников в процессе показа работ, баллы участника, в том числе в случае технических ошибок, могут быть изменены только в результате апелляции.

5.4. Рассмотрение апелляций участников олимпиады

Участник, не согласный с оцениванием его решений, имеет право подать апелляцию. Предметом апелляции является несоответствие выставленной оценки критериям оценивания решений, описанных в настоящих требованиях, методических материалах по проведению соответствующего этапа и условиях задач. Критерии и методика оценивания не могут быть предметом апелляции и пересмотру не подлежат.

Жюри устанавливает сроки и регламент подачи апелляций, однако срок, в течение которого могут быть поданы апелляции должен составлять не менее одного часа и должен завершиться не позднее третьего дня после олимпиады.

Основанием для проведения апелляции является заявление участника на имя председателя жюри, написанное по установленной форме.

Участник вправе требовать очного рассмотрения апелляции в его присутствии с использованием видеофиксации.

По результатам рассмотрения апелляции выносится одно из следующих решений:

- об отклонении апелляции и сохранении выставленных баллов;
- о частичном или полном удовлетворении апелляции и корректировке баллов.

Решение по каждой апелляции оформляется протоколом установленного вида, который подписывается членами жюри, принимавшими участие в рассмотрении апелляции. На основании протоколов рассмотрения апелляций вносятся соответствующие изменения в итоговые документы.

Окончательные итоги утверждаются жюри с учетом результатов рассмотрения апелляций и доводятся до сведения всех участников олимпиады.

6. Порядок подведения итогов олимпиады

6.1. Определение победителей и призёров

После рассмотрения апелляций жюри формирует рейтинги участников. Рейтинги формируются отдельно по классам. Участники в рейтинге упорядочиваются в порядке убывания их баллов. При равенстве баллов участники из одного класса в рейтинге указываются в алфавитном порядке, но считаются разделяющими одно и то же место.

Победители и призеры определяются отдельно по классам. Для этого жюри использует итоговые рейтинги.

Квота на общее количество победителей и призеров определяется организатором соответствующего этапа с учетом действующих нормативных документов. Следует обратить внимание, что порядок проведения всероссийской олимпиады не содержит дополнительных ограничений на количество баллов, которое должны набрать победители и призеры, в частности, ограничение в 50% набранных баллов, установленное пунктом 31 порядка проведения, относится только к заключительному этапу и не применяется на школьном и муниципальном этапе.

Для определения количества победителей и призеров по каждому классу квоту на общее количество победителей и призеров этапа рекомендуется распределять между классами пропорционально количеству участников из каждого класса. Жюри имеет право корректировать количество победителей и призеров этапа по каждому классу с

учетом баллов, набранных участниками из различных классов.

Списки победителей и призеров утверждаются организатором соответствующего этапа олимпиады. Победители и призеры муниципального этапа награждаются поощрительными грамотами.

6.2. Определение состава участников муниципального и регионального этапа

В соответствии с п. 46 порядка проведения всероссийской олимпиады школьников в муниципальном этапе принимают участие школьники, набравшие на школьном этапе необходимое число баллов, а также победители и призеры муниципального этапа предыдущего года.

Количество баллов, необходимое для участия в муниципальном этапе, устанавливается организатором муниципального этапа. Порядок проведения и настоящие рекомендации не устанавливают ограничений на проходные баллы. В частности, не следует устанавливать ограничение в 50% набранных на школьном этапе баллов для возможности участия в муниципальном этапе. Следует также обратить внимание, что статус призёра или победителя школьного этапа не влияет на возможность участия в муниципальном этапе. Например, проходные баллы могут быть установлены таким образом, что для участия в муниципальном этапе будут приглашены участники, не являющиеся призёрами школьного этапа.

В соответствии с п. 54 порядка проведения всероссийской олимпиады школьников в региональном этапе принимают участие школьники, набравшие на муниципальном этапе необходимое число баллов, а также победители и призеры регионального этапа предыдущего года.

Количество баллов, необходимое для участия в региональном этапе, устанавливается организатором регионального этапа. Порядок проведения и настоящие рекомендации не устанавливают ограничений на проходные баллы. В частности, не следует устанавливать ограничение в 50% набранных на муниципальном этапе баллов для возможности участия в региональном этапе. Следует также обратить внимание, что статус призёра или победителя муниципального этапа не влияет на возможность участия в региональном этапе. Например, проходные баллы могут быть установлены таким образом, что для участия в региональном этапе будут приглашены участники, не являющиеся призёрами муниципального этапа.

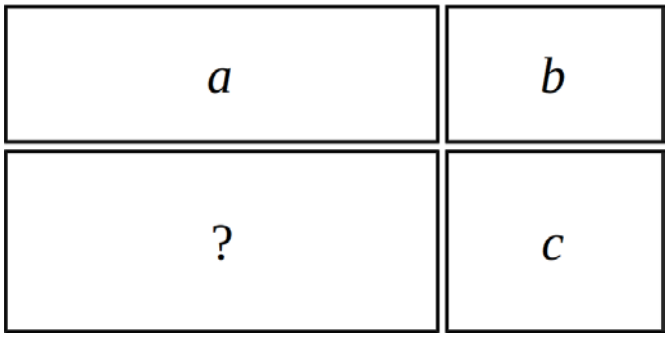
При установлении проходных баллов на муниципальный и региональный этап организаторам соответствующего этапа рекомендуется руководствоваться количеством участников предыдущего этапа и количеством рабочих мест, которые доступны для размещения участников соответствующего этапа.

В случае использования единого комплекта задач для нескольких классов, при определении проходных баллов на следующий этап олимпиады необходимо устанавливать проходные баллы для более младших классов не выше, чем для более старших классов.

Периметр (7-8 класс, компьютерная форма)

Условие

В здании был большой конференц-зал в форме прямоугольника. Его разделили на четыре меньших прямоугольных помещения, поставив две перпендикулярные стены (см.рисунок).



Для проведения ремонта необходимо определить периметр каждого из четырёх помещений. Три из четырёх помещений имеют периметр, равный a, b, c (в порядке обхода по часовой стрелке, начиная с левого верхнего угла плана). Определите периметр четвёртого помещения. Ответом на эту задачу является некоторое выражение, которое может содержать целые числа, переменные a, b и c (записываемые английскими буквами), операции сложения (обозначаются «+»), вычитания (обозначаются «−»), умножения (обозначаются «*»), деления (обозначаются «/») и круглые скобки для изменения порядка действий.

Запись вида «2a» для обозначения произведения числа 2 и переменной a неверная, нужно

писать «2*a».

Пример правильного по форме записи выражения: a + (b −c) * 2.

Ответ:

a + c - b

Критерии оценивания

При сдаче решения на проверку проверяющая программа проверяет, что выражение является корректным арифметическим выражением с использованием только разрешённых операций и переменных a, b, c, иначе решение получает статус ‒Неверный формат ответа‖.

При окончательной проверке любое арифметическое выражение, эквивалентное правильному ответу, оценивается в максимальный балл, например, выражение (a + b + c) - 2 * b также оценивается в максимальный балл. Для этого необходимо проверять эквивалентность двух выражений, для чего проверяющая программа может вычислять значения выражений на наборе различных значений a, b, c и проверять равенство полученных результатов.

Частичные баллы могут получать решения, содержащие некоторые ошибки,

например, решения вида a + b - c или b + c - a.

Крестраж (7-8 класс, компьютерная форма)

Условие

Волан де Морт спрятал один из крестражей в золотой рыбке. Эта рыбка живёт в пяти озёрах, соединённых между собой рекой. Озёра пронумерованы числами от 1 до 5, из озера 1 можно попасть в озеро 2, из озера 2 можно попасть в озёра 1 и 3 и т. д.

Гарри Поттер должен добыть эту золотую рыбку. Для этого у него есть волшебные червячки. Рыбка обязательно клюнет на наживку, если забросить её в озеро с рыбкой. Забрасывать наживку можно только в озеро. За один бросок можно бросить червячка только в одно озеро. Каждый волшебный червячок может быть использован только один раз. Если снасть с червячком забросили в озеро, а рыбки там не оказалось, то волшебная сила наживки исчезает и для следующей попытки требуется новый волшебный червячок. При этом рыбка чувствует Гарри Поттера и после каждого заброшенного червячка обязательно переплывает в одно из озёр, соседних с тем, в котором она находится. В самом начале рыбка может находиться в любом из пяти озёр.

Придумайте последовательность действий Гарри Поттера, при исполнении которой он обязательно поймает рыбку независимо от её первоначального место нахождения и дальнейших перемещений. В ответе нужно записать последовательность чисел через пробел
– номера озёр, в которые Гарри Поттер будет закидывать наживку, в том порядке, в котором он будет это делать. Чем меньше червячков потратит Гарри Поттер, тем больше баллов вы получите (при условии, что при исполнении вашего решения рыбка будет обязательно поймана).

Может показаться, что задача не имеет решения, но это не так. Рассмотрим случай трёх озёр. Гарри Поттер может закинуть наживку в озеро 2. Если он не поймает рыбку после этого, значит, она могла находиться в озёрах 1 или 3. После этого рыбка переплывает в соседнее озеро, и в каждом из этих случаев она попадёт в озеро 2. Поэтому вторую наживку Гарри Поттер снова закинет в озеро 2 и тогда обязательно поймает рыбку.

Ответ для трёх озёр: «2 2».

Ответ

Есть четыре наилучших решения:
2 3 4 2 3 4
2 3 4 4 3 2
4 3 2 2 3 4
4 3 2 4 3 2

Критерии оценивания

При сдаче решения на проверку проверяющая программа проверяет, что ответ представляет собой последовательность из чисел от 1 до 5, разделённых пробелами, иначе решение получает статус ―Неверный формат ответа‖.

При окончательной проверке проверяющая программа выполняет моделирование действий Гарри Поттера, определяя все возможные озёра, в которых может находиться рыбка после очередного хода, то есть проверяется, действительно ли указанная

последовательность действий Гарри Поттера позволяет всегда поймать рыбку, будем считать такие решения правильными.

Правильное решение, состоящее из 6 чисел, получает максимальный балл, другие правильные решения получают меньшее число баллов, в зависимости от длины ответа. Рекомендуется за любое правильное решение, независимо от его длины, давать не менее 30- 50% от максимального балла.

Также можно небольшим числом баллов оценивать решения, не являющиеся правильными, но позволяющие существенно сузить множество озёр, в которых может находиться рыбка, например, если после выполнения указанной последовательности действий рыбка может находиться только в одном каком-то озере.

Из разных цифр (7-8 класс, компьютерная форма)

Условие

Вам даны пять чисел:
4698
10000
123459876
987654321
9753102468
Для каждого из этих чисел найдите **минимальное** целое число, которое было бы **больше**
данного, и в записи которого все цифры были бы **различными.**

В ответе нужно записать пять целых чисел, записанных в отдельных строках. Порядок записи чисел в ответе менять нельзя. Если вы не можете найти ответ для какого-то из данных чисел, вместо этого ответа запишите любое целое число.

Ответ

4701
10234
123460578
1023456789
9753102486

Критерии оценивания

Задача разбивается на пять отдельных примеров, демонстрирующих все особенности алгоритма построения нужного числа. Каждый пример оценивается отдельно.

При сдаче решения на проверку проверяющая программа проверяет, что ответ представляет собой пять чисел, записанных в пяти разных строках, иначе решение получает статус ‒Неверный формат ответа‖.

При окончательной проверке проверяющая программа оценивает каждый правильный ответ из пяти определённым числом баллов независимо от остальных тестов. Балл за задачу складывается из суммы баллов за правильные ответы на

примеры.

Гирьки (7-8 класс, компьютерная форма)

Условие

У ювелира есть весы с двумя чашками, он может определять равны ли массы грузов, лежащих на двух чашках, а если не равны – то на какой чашке лежит более легкий груз.

Масса ювелирного изделия, которую нужно определить ювелиру, является целым числом от 1 до 25 грамм. Ювелир должен запасти набор гирек (их массы также должны быть целыми числами), используя которые он может определить любую возможную целочисленную массу от 1 до 25 грамм. Для определения массы ювелир может производить любое число взвешиваний, может использовать все или только часть набора гирек, может класть гирьки на разные чашки весов и т.д. Определите набор гирек, содержащий минимальное возможное число гирек, используя который можно определить любую возможную целочисленную массу от 1 до 25.

В ответе нужно записать массы гирек в подготовленном наборе через пробел. За правильный набор из трёх гирек вы получите 10 баллов, из четырёх гирек – 5 баллов, из пяти гирек – 2 балла.

Ответ

2 6 18

Критерии оценивания

При сдаче решения на проверку проверяющая программа проверяет, что ответ представляет собой последовательность чисел, записанных через пробел, иначе решение получает статус ―Неверный формат ответа‖. Правильность приведённого ответа не проверяется.

При окончательной проверке проверяющая программа проверяет, действительно ли этот набор удовлетворяет условию задачи. Для этого перебираются все возможные массы от 1 до 25 и для каждой массы перебираются все возможные результаты взвешиваний, для различного размещения указанных гирек на двух чашках весов. Каждая гирька может находиться на одной чашке с грузом, на другой чашке или не участвовать во взвешивании.

Если существуют две какие-то массы, для которых результаты всех взвешиваний будут одинаковыми, то эти массы будут неразличимы, значит, набор будет неподходящим.

Правильное решение из 3 гирек оценивается в 10 баллов, правильное решение из 4 гирек (например, ‒1 3 9 18‖) оценивается в 5 баллов, решение из 5 гирек (например, 1 2 4 8 16) оценивается в 2 балла.

Два подарка (9-11 класс, компьютерная форма)

Условие

Сеня выбирает себе подарки на новый год. Он знает, что Дед Мороз купит ему ровно два подарка: один якобы от мамы, а другой якобы от папы.

В магазине, где Дед Мороз будет покупать подарки, продаются n подарков, про каждый подарок известна его цена: цена i-го подарка равна a_i рублей.

Сеня знает, что Дед Мороз может потратить на покупку его подарков не больше x рублей. Разумеется, он хочет получить как можно более дорогие подарки. Таким образом, он хочет выбрать два различных подарка с максимальной суммарной ценой, но при этом она не должна превышать x.

Помогите Сене выбрать себе подарки.

Формат входных данных

Первая строка ввода содержит два целых числа: n и x ($2 \leq n \leq 100000$, $2 \leq x \leq 10^9$).

Вторая строка ввода содержит n целых чисел: a_1, a_2, ..., a_n ($1 \leq a_i \leq 10^9$).

Гарантируется, что существует два подарка с суммарной ценой не больше x.

Формат выходных данных

Выведите одно целое число: максимальную суммарную цену двух различных подарков, не превышающую x.

Пример

Ввод	**Вывод**
6 18 5 3 10 2 4 9	15

Число делителей (9-11 класс, компьютерная форма)

Условие

Задано число n. Требуется найти число от 1 до n, включительно, которое имеет максимальное число положительных целых делителей. Например, если n = 20, то искомое число — 12, у него 6 делителей: 1, 2, 3, 4, 6 и 12.

Формат входных данных

На вход подается одно число n ($1 \leq n \leq 100000$).

Формат выходных данных

Выведите на первой строке число от 1 до n, включительно, которое имеет максимальное число делителей. На второй строке выведите число его делителей. Если есть несколько чисел от 1 до n с максимальным числом делителей, выведите любое из них.

Пример

Ввод	Вывод
20	12 6

Решение

Решение на 56 баллов.

Для каждого числа от 1 до n найдём количество его делителей. Для нахождения количества делителей числа x, перебираем все числа от 1 до x и проверяем, делится ли x на него. Данное решение имеет сложность $O(n^2)$.

Решение на 94 балла.

Предыдущее решение можно ускорить, если заметить, что для нахождения количества делителей числа x, можно перебирать только числа до квадратного корня из x.

Решение на 100 баллов.

Заведём массив d. Будем перебирать числа от 1 до n. Пусть сейчас рассматривается число x. Для каждого числа k, такого что $kx \leq n$, прибавляем к d[kx] единицу. Чтобы найти ответ на задачу нам нужно просто найти максимум в этом массиве

Родительский совет (9-11 класс, компьютерная форма)

Условие

В управляющий совет школы входят родители, учителя и учащиеся школы, причём родителей должно быть не менее одной трети от общего числа членов совета. В настоящий момент в совет входит N человек, из них K родителей. Определите, сколько родителей нужно дополнительно ввести в совет, чтобы их число стало составлять не менее трети от числа членов совета.

Формат входных данных

Программа получает на вход два целых числа N и K ($N > 0$, $0 \leq K \leq N$), записанные в отдельных строках, — текущее число членов совета и число родителей в совете.

Формат выходных данных

Программа должна вывести единственное число — минимальное число родителей, которое необходимо ввести в совет.

Ограничения и система оценивания

Решение, правильно работающее в случае, когда числа N и K не превосходят 100,

будет оцениваться в 60 баллов.

Решение, правильно работающее в случае, когда числа N и K не превосходят 2·109, будет оцениваться в 100 баллов.

Пример

Ввод	**Вывод**
27 7	3

Счастливые билеты (9-11 класс, компьютерная форма)

Условие

На автобусных билетах указываются их номера. Номера всех билетов всегда записываются при помощи одного и того же количества цифр, при этом число используемых цифр чётно. При необходимости числа дополняются ведущими нулями. К примеру, если для записи используют 4 цифры, то 514 будет записано как 0514. Билеты отпечатаны на лентах, билеты на каждой ленте нумеруются подряд числами от 00...01 до 99...99.

Счастливым считается тот билет, у которого сумма цифр первой половины равна сумме цифр второй половины, например, билеты 1001 и 123051 счастливые, а 7778 и 39 – нет.

Сегодня Дима зашел в автобус, и кондуктор выдал ему билет с номером N. Поскольку Диме ехать достаточно долго, а заняться чем-нибудь надо, он стал думать, какой номер будет иметь следующий счастливый билет, выданный из той же ленты, что и Димин билет. Если в текущей ленте не осталось счастливых билетов, Диму интересует номер минимального счастливого билета из новой ленты.

В первой и единственной строке входного файла содержится номер Диминого билета N, записанный с ведущими нулями. Количество цифр в записи числа N не превосходит 100 000 и чётно.

Программа должна вывести номер следующего счастливого билета из текущей ленты в таком же формате. Если такого билета не существует, надо вывести номер минимального счастливого билета из новой ленты. В выводе не должно быть пробелов, пустых строк в начале вывода.

Пример

Ввод	Вывод
0514	0523

Диме был выдан счастливый билет (сумма цифр обеих половин равна 5), но Диму не интересует номер его билета, его интересует номер следующего счастливого билета.

Система оценивания

Решение, правильно работающее только для случаев, когда номер билета содержит ровно 4 цифры, будет оцениваться в 20 баллов.

Решение, правильно работающее только для случаев, когда номер билета содержит ровно 8 цифр, будет оцениваться в 20 баллов (вместе с предыдущей группой – 40 баллов).

Решение, правильно работающее только для случаев, когда номер билета содержит не более 16 цифр, будет оцениваться в 60 баллов.

Приложение 2. Методические рекомендации по разработке материалов задач для решения с использованием универсальных языков программирования

Подготовка условия

1. Все, не относящееся собственно к постановке задачи - предыстория, легенда и т. п. - должно находиться не более чем в одном абзаце. Этот абзац должен идти первым. В дальнейшем допускается иногда вставлять мотивирующие предложения, связанные с легендой, но не более одного подряд, и в целом их должно быть как можно меньше.
2. Легенда должна вводить мотивацию в постановку задачи, но не затуманить ее и не вводить в заблуждение. Желательно, чтобы легенда не содержала отдельными предложениями сведений, не требующихся для постановки задачи.
3. Условие задачи должно быть последовательным и четким. Никакая фраза не должна допускать неоднозначного трактования. Термины и определения можно использовать только после их введения. По мере чтения условия у участника должна последовательно складываться картина того, что требуется сделать.
4. Следует использовать простые и понятные фразы, избегать витиеватостей и длинных сложноподчиненных предложений.
5. Условие задачи должно быть грамотным и не должно использовать просторечных выражений.
6. Не допускаются сокращения, кроме ―и т.п.‖ и ―и т.д.‖ (а эти выражения не рекомендуется использовать в условиях). Следует писать полностью ―то есть‖, ―так как‖.
7. Последний абзац условия должен резюмировать условие и еще раз четко формулировать, что требуется сделать.
8. Для всех задач соревнования рекомендуется выбрать единый стиль - либо безличного обращения (―требуется найти‖, ―требуется вывести‖), либо личного (―найдите‖,
   ―выведите‖). В любом случае, в рамках одного условия точно должен быть единый стиль.
9. Раздел ―Формат входных данных‖ должен содержать формат входных данных и ограничения. Он не должен пояснять задачу или вводить дополнительные условия, кроме числовых ограничений на входные данные. Прочие ограничения на входные данные (например, возрастание массива) должны быть также прописаны в основном условии (хотя и должны быть повторены еще раз в разделе ―Формат входных данных‖).
10. Раздел ―Формат выходных данных‖ должен содержать формат выходных данных. В нем также можно еще раз повторить, что требуется найти.

Особенности при подготовке условия в системе верстки TeX

11. Формулы должны быть заключены в символы доллара. Одиночные

переменные, которые обозначают математические объекты являются формулами. Буквы, которые не

обозначают математические объекты - не являются формулами. Например,

У Пети n поросят - ОК
У Пети n поросят - неправильно

Дана строка s. - ОК
Дана строка s - неправильно

На кольцевой дороге города N построили развязку - ОК
На кольцевой дороге города N построили развязку - неправильно

12. Знаки препинания, которые относятся к формуле должны быть включены в формулу. Знаки препинания, которые относятся к предложению, не должны быть включены в формулу, например:

Заданы целые числа m, n и k - ОК.
Заданы целые числа m, n и k -
неправильно.

Задано целое число n ($1 \le n \le 100$) - ОК.
Задано целое число n $(1 \le n \le 100)$ - неправильно.

Площадь трапеции равна $(a + b)\cdot h / 2$ - ОК
Площадь трапеции равна ($a + b$)$\cdot h / 2$ - неправильно

Задана последовательность a_1, a_2, …, a_n -
неправильно. Задана последовательность $a_1, a_2, \ldots, a_n$ - ОК.

13. Не используйте программистские обозначения в формулах, используйте математические.

Выведите $2n$ чисел - ОК
Выведите $2\times n$ чисел - ОК (хотя в этом конкретном примере \times не нужен) Выведите $2\cdot n$ чисел - ОК (хотя в этом конкретном примере \cdot не нужен) Выведите $2*n$ чисел - неправильно

―Исключающее или‖ двух чисел обозначается $x \oplus y$ - ОК

14. Строковые литералы следует набирать моноширинным шрифтом, а не формулой и не просто так. Кавычки должны быть русскими << >> в русских условиях и английскими направленными `` _‘ в английских фразах. Двойную кавычку (символ с кодом 34) не использовать. Кавычки моноширинными не делать. Например,

Выведите в выходной файл <<\texttt{Impossible}>> - ОК
Выведите в выходной файл \texttt{<<Impossible>>} -
неправильно Выведите в выходной файл <<$Impossible$>> -
неправильно Выведите в выходной файл <<Impossible>> -

неправильно

15. Фрагменты текста, не являющиеся формулами, не следует делать формулами. Например,

    В XXI веке изобрели телепорт - ОК
    В XXI веке изобрели телепорт - неправильно

16. Одиночные числа не следует делать формулами. Например,

    В 1961 году Юрий Гагарин полетел в космос - ОК
    В 1961 году Юрий Гагарин полетел в космос - неправильно
17. Числительные от 1 до 10 обычно пишутся текстом. Большие - числом. Например,

    У Васи было три поросенка - ОК
    У Васи было 3 поросенка - неправильно

    У Пети было три тысячи пятьсот двенадцать поросят -
    неправильно У Пети было 3512 поросят - ОК
18. Порядковые числительные с параметром, либо большие 10, пишутся с суффиксом --й‖ (--я‖) и аналогично склоняются (первая гласная суффикса опускается). Например,

    Выведите k в лексикографическом порядке строку -
    неправильно Выведите k-ю в лексикографическом порядке
    строку - ОК
    Выведите k-ую в лексикографическом порядке строку -
    неправильно Выведите k-тую в лексикографическом порядке
    строку - неправильно Ошибка была в 112-й строке - ОК
19. Форматирование должно быть только высокоуровневым и логическим. Не разрешается использовать низкоуровневое форматирование (задавать размеры в сантиметрах/пикселях и т.п.) либо применять форматирование не по назначению (например, использовать \big для создания заголовков итп).
20. В качестве тире следует использовать три минуса: ---. Перед тире следует ставить неразрывный пробел. Обратите внимание, что перенос строки или пробел перед неразрывным пробелом уничтожают его неразрывность. Также можно использовать обозначение для тире ---- (двойная кавычка и затем три минуса), в этом случае перед тире ставится пробел. Например,

    Нептун - восьмая планета Солнечной системы -
    неправильно Нептун -- восьмая планета Солнечной
    системы - неправильно Нептун --- восьмая планета
    Солнечной системы - неправильно Нептун~--- восьмая
    планета Солнечной системы - ОК
    Нептун ---- восьмая планета Солнечной системы - ОК
    Нептун ~--- восьмая планета Солнечной системы - неправильно
21. Ограничения на численные значения параметров в формате входных данных пишутся в том же предложении, что и описание места этих параметров во входных данных, -
    в скобках в конце.

    В первой строке входных данных находится целое число n ---- количество городов ($1 \le n \le 100$). - ОК
    В первой строке входных данных находится целое число n ($1 \le n \le 100$) --

-- количество городов. - неправильно

22. Если вы задаете ограничение сразу на несколько переменных, пишите их через запятую. В этом случае, если у вас подряд идет несколько блоков ограничений, их следует разделять знаком точки с запятой.

    В первой строке входных данных находятся целые числа a, b и c –--- количество

городов, сел и деревень, соответственно ($1 \le a, b \le 100$; $1 \le c \le 1000$). - ОК
В первой строке входных данных находятся целые числа a, b и c –--- количество городов, сел и деревень, соответственно ($1 \le a, b \le 100$, $1 \le c \le 1000$). - плохо, запятая играет разную роль
В первой строке входных данных находятся целые числа a, b и c –--- количество городов, сел и деревень, соответственно ($1 \le a \le 100$, $1 \le b \le 100$, $1 \le c \le 1000$). - допустимо, хотя чем больше блоков ограничений, тем тяжелее воспринимается.

23. Всегда ставьте пробел перед скобкой в предложении.

    Это условие понятное (мы надеемся, что так и есть). - ОК
    Это условие понятное(мы надеемся, что так и есть). - неправильно

    Во второй строке находится число n ($1 \le n \le 100$). - ОК
    Во второй строке находится число n($1 \le n \le 100$). - неправильно

Примеры в условии

24. Примеры необходимо подбирать таким образом, чтобы они проясняли потенциально менее понятные фрагменты условия, демонстрировали особенности ввода и вывода.
25. Ответ на пример необходимо получить вручную. Если этот процесс нетривиальный, то следует написать пояснение к примеру или добавить картинку.
26. Если решение жюри выводит другой ответ на пример, то следует проверить ответ с использованием проверяющей программы, чтобы убедиться, что ответ в условии правильный.
27. Лучше подбирать примеры на все возможные случаи в решении, кроме варианта, когда одна из целей задачи - догадаться до того, что такой случай бывает.
28. Примеров не должно быть слишком много.

Выбор ограничений и написание решения

29. По каждой задаче должно быть решение на языке Pascal, Python, C++ или Java, которые написаны естественным образом без неасимптотических оптимизаций (например, быстрого ввода) и укладываются в TL с двухкратным запасом.
30. Если большие ограничения на размер ввода не являются необходимыми для отсечения неэффективных алгоритмов, следует делать достаточно маленькие ограничения, чтобы программы на Python легко укладывались в TL.

Написание проверяющей программы

31. Проверяющую программу рекомендуется писать на C++ с использованием библиотеки testlib (https://github.com/MikeMirzayanov/testlib).
32. В целом рекомендуется использование стандартных проверяющих программ из поставки testlib для C++ и/или встроенных в Polygon.
33. Проверяющая программа не должна предполагать ничего о том, что выведут участники. Все должно проверяться. В частности (но не только!)
    - Если вы хотите создать массив/вектор размера, который вы прочитали из выходного файла участника, проверьте его на корректность.
    - Если вы хотите обратиться по индексу в массив, а индекс вы прочитали из выходного файла участника, проверьте его на корректность.
    - Если вы хотите делать операции с числами, которые вы прочитали из входного файла участника, убедитесь, что у вас не будет переполнения.
    - Если вы прочитали из выходного файла строку, которая по *условию* должна удовлетворять некоторым *условиям*, прежде чем это использовать - проверьте это.

Подготовка тестов

34. Первые несколько тестов должны совпадать с тестами из условия.
35. Большие тесты необходимо сгенерировать, генератор тестов можно, например, писать на C++ с использованием библиотеки testlib.
36. Тесты должны быть корректными текстовыми файлами. Каждая строка, включая последнюю, должна завершаться переводом строки.
37. Тестирование может проводиться как под Windows, так и под Linux. Перевод строки под Windows задается двумя символами: 13 и 10 в этом порядке. Перевод строки
    под Linux задается одним символом с кодом 10. При генерации под Windows должны получаться файлы с Windows-переводами строк, а при генерации под Linux - файлы с Linux-переводами строк.
    - В программах на C++ ―<< endl‖ и –\n‖ в ―cout << – и ―printf‖ выводят правильно. Специально выводить –\r‖ не надо!
    - В программах на Java println выводит правильно. Если вы выводите с помощью printf, то надо выводить –%n‖, а не –\n‖ .
    - В программах на Python print выводит одну строку правильно, write выводит правильно, если вы пишете –\n‖. Если вы используете print для вывода нескольких строк, при генерации под Windows то надо писать –\r\n‖, не используйте print для вывода более чем одной строки.

38. Если иное не оговорено явно в условии задачи, тесты должны удовлетворять следующим условиям:
    - В строках не должно быть пробелов в начале или в конце
    - В тестах не должно быть пустых строк, в том числе в конце файла.
    - В тестах не должно быть двух пробелов подряд
    - В тестах не должно быть символов с кодами меньше 32, кроме переводов строк, и символов с кодами больше 126.
39. Данные во входном файле должны быть разбиты на строки в точности так, как описано в условии задачи. Лишних данных в тестах быть не должно.
40. Генератор тестов должен быть детерминированным. Он должен выдавать одни и те же тесты при повторных запусках.
41. Рекомендуется использовать ровно один из двух подходов: ―один запуск - один тест‖ - генератор выводит ровно один тест на свой стандартный вывод ИЛИ ―один генератор, все тесты‖ - генератор выводит все тесты в файлы {номер_теста} в текущий каталог. Во втором случае не следует использовать ручные тесты.
42. Тесты должны по возможности покрывать все крайние случаи, в частности, содержать минимальные и максимальные подходящие под ограничения входные данные, крайние и особые случаи. Не рекомендуется ограничиваться случайными тестами.

Написание валидаторов

43. Для избежания ошибок при подготовке тестов рекомендуется использовать валидаторы
    - специальные программы, проверяющие корректность тестов.
44. Валидатор может быть написан на любом языке программирования. Если вы готовите задачи не в Polygon, то скрипт генерации тестов должен также компилировать
    и запускать валидатор.
45. Валидатор принимает на стандартный вход тест и выходит с кодом 0, если тест корректный, иначе выходит с ненулевым кодом. При этом в стандартный вывод он может написать описание ошибки.
46. Для написания валидаторов можно применять библиотеку testlib.

Приложение 3. Рекомендуемые ресурсы интернет для скачивания и установки программного обеспечения

Программное обеспечение, рекомендуемое для использования на олимпиаде, размещается на следующих сайтах:

- MinGW GNU C++ – https://sourceforge.net/projects/mingw-w64/
- Free Pascal – https://www.freepascal.org/
- Microsoft Visual C++, C#, Basic – https://visualstudio.microsoft.com/vs/express/
- Oracle Java – https://www.oracle.com/technetwork/java/index.html
- OpenJDK Java – https://jdk.java.net/12/
- Python – https://www.python.org/
- Pascal ABC – http://pascalabc.net/
- Free Basic – https://www.freebasic.net/
- Code::Blocks – http://www.codeblocks.org/
- IntelliJ IDEA – https://www.jetbrains.com/idea/
- PyCharm – https://www.jetbrains.com/pycharm/
- CLion – https://www.jetbrains.com/clion/
- Wing IDE – https://wingware.com/
- Sublime Text – https://www.sublimetext.com/
- Vim – https://www.vim.org/
- Far Manager – https://www.farmanager.com/
- Geany – https://www.geany.org/

Для доступа участников к документации рекомендуется разместить на компьютерах участников или в локальной сети локальные копии:

- документации по языку C++, например http://cppreference.com;
- документации по языку Free Pascal с https://www.freepascal.org/docs.var;
- документации по Java API с https://docs.oracle.com/en/java/;
- документации по языку Python с https://docs.python.org/3/;
- документации по другим доступным языкам программирования.

Приложение 4. Ссылки на страницы школьного и муниципального этапа некоторых регионов

Москва https://olympiads.ru/moscow/

Санкт-Петербург http://neerc.ifmo.ru/school/spb/municipal.html

Московская область https://mipt.ru/abiturs/olympiads/vos/informatics/subreg1819/

Приложение 5. Контакты для консультаций с ЦПМК