Утверждены на заседании центральной предметно-методической комиссии всероссийской олимпиады школьников по информатике 18.06.2024 г. (Протокол № 4)

Методические рекомендации по проведению школьного и муниципального этапов всероссийской олимпиады школьников по информатике в 2024/25 учебном году

СОДЕРЖАНИЕ

Введение……..3
1. Специфика проведения олимпиады по информатике…….4
1.1. Основные принципы 4
1.2. Особенности показа работ 5
1.3. Особенности процедуры апелляции 5
2. Принципы формирования комплектов олимпиадных заданий школьного и муниципального этапа олимпиады……...6
2.1. Школьный этап для учащихся 5-6 классов 6
2.2. Школьный и муниципальный этапы для учащихся 7-8 классов 7
2.3. Школьный и муниципальный этапы для учащихся 9-11 классов 8
3. Методические подходы к составлению заданий школьного и муниципального этапа олимпиады……...9
3.1. Задания для проведения тура в бланковой форме..... 9
3.2. Задания в компьютерной форме с кратким ответом 10
3.3. Задания на использование компьютерных сред для формальных исполнителей или виртуальных лабораторий 14
3.4. Задания по программированию для решения с использованием универсальных языков.15
4. Необходимое материально-техническое обеспечение для выполнения олимпиадных заданий школьного и муниципального этапа олимпиады……21
4.1 Материально-техническое обеспечение при использовании заданий в бланковой форме 21
4.2. Материально-техническое обеспечение при компьютерной форме проведения этапа 21
5. Перечень справочных материалов, средств связи и электронно-вычислительной техники, разрешенных к использованию во время проведения олимпиады……..22
6. Критерии и методика оценивания выполненных олимпиадных заданий……...23
7. Использование учебной литературы и интернет-ресурсов при подготовке школьников к олимпиаде……..23
Приложения……...……...24

Введение

Настоящие рекомендации по организации и проведению школьного и муниципального этапов всероссийской олимпиады школьников (далее – олимпиада) по информатике составлены в соответствии с Порядком проведения всероссийской олимпиады школьников, утвержденным приказом Министерства просвещения Российской Федерации от 27 ноября 2020 г. № 678 «Об утверждении Порядка проведения всероссийской олимпиады школьников».

Олимпиада по информатике проводится в целях выявления и развития у обучающихся творческих способностей и интереса к научной (научно-исследовательской) деятельности, пропаганды научных знаний.

Олимпиада проводится на территории Российской Федерации.

Рабочим языком проведения олимпиады является русский язык.

Участие в олимпиаде индивидуальное, олимпиадные задания выполняются участником самостоятельно, без помощи посторонних лиц.

Сроки окончания этапов олимпиады: школьного этапа олимпиады – не позднее 01 ноября; муниципального этапа олимпиады – не позднее 25 декабря.

Школьный этап олимпиады проводится по заданиям, разработанным для 5-11 классов, муниципальный – для 7-11 классов. Участник олимпиады выполняет по своему выбору олимпиадные задания, разработанные для класса, программу которого он осваивает, или для более старших классов. В случае прохождения участников олимпиады, выполнивших задания, разработанные для более старших классов по отношению к тем классам, программы которых они осваивают, на следующий этап олимпиады указанные участники олимпиады и на следующих этапах олимпиады выполняют олимпиадные задания, разработанные для класса, который они выбрали на предыдущем этапе олимпиады.

Методические рекомендации включают: методические подходы к составлению олимпиадных заданий школьного и муниципального этапов олимпиады; принципы формирования комплектов олимпиадных заданий; необходимое материально-техническое обеспечение для выполнения олимпиадных заданий; перечень справочных материалов, средств связи и электронно-вычислительной техники, разрешенных к использованию во время проведения олимпиады; критерии и методику оценивания выполненных олимпиадных заданий.

Дополнительную информацию по представленным методическим материалам можно получить по электронной почте, обратившись по адресу: **regional.roi@gmail.com** в центральную предметно-методическую комиссию всероссийской олимпиады школьников по информатике (далее – ЦПМК).

1. Специфика проведения олимпиады по информатике

1.1. Основные принципы

Олимпиада по информатике обычно проводится с использованием компьютеров. Тем не менее при проведении школьного этапа олимпиады для 5-6 классов допускается проведение в бланковой форме, когда участникам предлагаются задания с развёрнутым ответом, решения которых записываются на бумаге с последующей проверкой жюри.

Для автоматизации проверки заданий обычно используется *тестирующая система*. Участники с использованием специального интерфейса отправляют ответы на задания либо программы-решения на проверку во время тура и получают информацию о корректности своего решения в соответствии с процедурами, описанными далее в настоящих рекомендациях.

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

В случае использования компьютеров для проведения этапа перед началом каждого тура все компьютеры участников должны находиться во включённом 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 олимпиады. 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 возможны только в сопровождении дежурного.

При контроле времени тестирующей системой приём решений автоматически прекращается, отправка решений в тестирующую систему после окончания тура невозможна.

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 произошёл не по вине участника.

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

1.2. Особенности показа работ

В случае использования онлайн-тестирования, при котором результаты проверки решений сообщаются участникам олимпиады 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 зачёта после окончания тура.

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

1.3. Особенности процедуры апелляции

Участник, не согласный с оцениванием его решений, имеет право подать апелляцию. Предметом апелляции является несоответствие выставленной оценки критериям оценивания решений. Содержание заданий, критерии и методика оценивания не могут быть предметом апелляции и пересмотру не подлежат. В частности, предметом апелляции не может быть распределение баллов за какие-то конкретные тесты, частные случаи решений и т. д.

Предметом апелляции в задачах по программированию может быть:

- несоответствие тестов условию задачи;
- несоответствие тестов ограничениям на подзадачи;
- некорректная работа проверяющей программы, т. е. правильный вывод решения участника олимпиады засчитывается как неправильный.

Оргкомитет устанавливает сроки и регламент подачи апелляций, однако срок, в течение которого могут быть поданы апелляции, должен составлять не менее одного часа.

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

По результатам рассмотрения апелляции выносится одно из следующих решений:

- отклонить апелляцию, сохранив количество баллов;

– удовлетворить апелляцию с понижением количества баллов;

– удовлетворить апелляцию с повышением количества баллов.

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

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

2. Принципы формирования комплектов олимпиадных заданий школьного и муниципального этапа олимпиады

2.1. Школьный этап для учащихся 5-6 классов

Для учащихся 5-6 классов проводится только школьный этап олимпиады.

Рекомендуется проведение олимпиады в один тур, продолжительность тура от 45 до 90 минут. При наличии задач по программированию или заданий на составление алгоритмов в компьютерной среде исполнителя продолжительность тура может быть увеличена до 120 минут.

Школьный этап олимпиады для 5-6 классов может проводиться в одной из следующих форм или с использованием заданий нескольких форм:

– бланковая форма – предлагаются задания с развёрнутым ответом, решения которых записываются на бумаге с последующей проверкой жюри школьного этапа на основании критериев, разработанных соответствующей предметно-методической комиссией;

– компьютерная форма заданий с кратким ответом – задания, ответ на которые записывается в виде одного или нескольких чисел, одной или нескольких строк текста, с вводом ответа в тестирующую систему и с последующей автоматической проверкой ответа;

– задания на использование компьютерных сред для формальных исполнителей или виртуальных лабораторий – задания выполняются в учебной среде, проверка заданий может быть автоматической или ручной.

Задания, требующие навыков использования какой-либо конкретной учебной среды программирования (например, Scratch или Логомиры), могут предлагаться на школьном этапе по решению соответствующей предметно-методической комиссии, только если во всех образовательных организациях данного муниципального образования созданы условия для изучения данной среды, т. е. такие задания должны быть доступны всем обучающимся.

Не рекомендуется предлагать задания по программированию с использованием универсальных языков, таких, как Python, C++, Pascal, Java, C#, но при наличии в регионе

большого числа учащихся 5-6 классов, владеющих навыками программирования, задания школьного этапа могут включать несколько таких заданий.

Рекомендуется включать в вариант 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 школьного этапа олимпиады.

2.2. Школьный и муниципальный этапы для учащихся 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 https://contest.yandex.ru, Codeforces https://codeforces.com, Ejudge http://ejudge.ru, тестирующей системы ОЦ «Сириус» https://uts.sirius.online 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 Python, C++, Pascal, 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 включать как задания 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-методических комиссий, только если во всех образовательных организациях 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 этапа 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, как задания 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.

2.3. Школьный и муниципальный этапы для учащихся 9-11 классов

Для учащихся 9-11 классов проводятся школьный и муниципальный этапы олимпиады. Далее участники муниципального этапа, 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 учётом 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 Python, C++, Pascal, 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, с одной стороны, 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 должно 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. Методические подходы к составлению заданий школьного и муниципального этапа олимпиады

3.1. Задания для проведения тура в бланковой форме

3.1.1. Принципы составления заданий

Задания в бланковой форме могут предлагаться на школьном этапе учащимся 5-6 классов. Задания предполагают запись решения в форме с развёрнутым ответом; проверка заданий осуществляется членами жюри. Если критерии оценивания какого-либо задания предусматривают снижение баллов за отсутствие обоснования ответа, в условии задания должно быть указано: «Обоснуйте полученный ответ». Желательно включение задач, в которых возможно получение различных верных ответов с возможностью оценивания их эффективности: например, длина пути, пройденного исполнителем, количество команд, использованных для составления алгоритма, количество гирек, использованных для решения задачи, и т. д. В условиях таких задач должно быть указание на то, что необходимо получить наилучший ответ, например, в виде «Постарайтесь составить алгоритм, содержащий наименьшее число команд» или «Желательно использовать как можно меньше гирек» и т. д.

3.1.2. Тематика заданий

Примерные темы заданий бланковой формы для 5-6 классов:

- логические задачи;
- комбинаторные задачи;
- задачи на сортировки, взвешивания, перекладывания, переливания, переправы;

– лабиринтные задачи;

– составление алгоритмов для исполнителя;

– выигрышные стратегии для простейших игр.

3.1.3. Критерии и методики оценивания

Жюри олимпиады проверяет выполненные задания в соответствии с критериями, разработанными предметно-методическими комиссиями. Все задания оцениваются одинаковым максимальным числом баллов. Критерии оценивания заданий должны предусматривать выставление частичного балла за решения, по каждой задаче должна быть составлена шкала оценивания решений задачи. Возможные подходы к составлению такой шкалы:

– если задача предусматривает обоснование полученного ответа, то баллы могут снижаться за отсутствие такого обоснования, наличие ошибок в доказательстве, рассмотрение только отдельных частных случаев и т. д. При этом оценка не может снижаться за сложность, запутанность или большой объем приведенного решения в случае его полноты и корректности;

– если задание предусматривает нахождение ответа разной эффективности (количество команд в алгоритме, количество операций при переливаниях, количество использованных гирек для взвешивания, длина пройденного исполнителем пути и т. д.), то баллы выставляются в зависимости от эффективности найденного ответа (максимальный балл выставляется за наилучшее возможное решение, далее баллы снижаются в зависимости от эффективности найденного ответа. За любое решение, без требований к его эффективности, рекомендуется выставлять 25-50 % от максимального балла).

Задача может разбиваться на несколько отдельных пунктов, подзадач или примеров, при этом каждый пункт оценивается отдельно. Баллы за всю задачу разбиваются на баллы за отдельные пункты.

3.2. Задания в компьютерной форме с кратким ответом

3.2.1. Принципы составления заданий

Задания в компьютерной форме с кратким ответом представляют собой задания, ответ на которые 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.

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 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 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.;

– 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;

– 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.

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). Если для выполнения заданий необходимо какое-либо специальное программное обеспечение, оно также должно быть установлено.

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 «Неправильный формат записи ответа». В этом случае желательна выдача дополнительного комментария тестирующей системы о несоответствии сданного ответа формату, описанному 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, длины найденного маршрута, количества 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, управления 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.

Для выполнения заданий на обработку данных в формате электронных таблиц, на компьютерах должно быть установлено необходимое программное обеспечение (например, Microsoft Excel или Libre Office Calc).

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 формирует список языков программирования, доступных для решения задач. В список рекомендуется включить распространённые языки программирования общего назначения, в том числе:

- Python;
- C++;
- Pascal;
- 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: КуМир, Kotlin, Rust, C, D и др.

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 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 (за исключением наиболее трудных задач в варианте, если входные данные в этих задачах устроены сложным образом, например, даны N пар чисел, в этом случае допускается во входных данных задать число N, затем N строк, содержащих по два числа, разделённых пробелом). 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. На школьном и муниципальном этапе не рекомендуется включать в условие длинную легенду или мотивацию, лучше стараться делать короткие относительно формальные условия.

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 или возможное значение целых чисел в выводе верного решения задачи 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.

Все задачи школьного и муниципального этапов должны решаться на полный балл на наиболее распространённых языках программирования (Python и C++).

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 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.
– Задания, 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 определённым в условии задачи именем, размещённого 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, 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 условия задачи.

Во время тура участнику может сообщаться информация о результатах проверки его решения. Возможные формы сообщаемой информации:

– участнику сообщаются баллы, набранные решением, и результат проверки решения на каждом тесте («OK», «Неправильный ответ», «Превышено максимальное время работы» и т. д.). Сами тесты при этом не сообщаются;

– участнику сообщаются только баллы, набранные решением. Информация о прохождении отдельных тестов не сообщаются;

– участнику сообщается информация о прохождении только части тестов. Об этих тестах может сообщаться как детальная информация, так и только количество набранных баллов. О результатах проверки на остальных тестах участнику информация не предоставляется. Типичным вариантом использования такой частичной информации является случай, когда максимальные по ограничениям тесты (предполагающие реализацию наиболее эффективного решения) выделяются в группу, оцениваемую в 30-60% от полного балла, и результат проверки на этой группе является скрытым от участника, а открытым является результат проверки на тестах, не требующих наиболее эффективного решения;

– участнику сообщается только результат проверки на тестах из условия (оцениваемых в 0 баллов).

Допускается сочетание разных форм сообщаемой информации о результатах проверки в разных задачах.

На школьном и муниципальном этапе рекомендуется сообщать информацию о результатах проверки во время тура. При потестовой оценке рекомендуется указывать результат проверки на каждом тесте или баллы за каждую подзадачу. При оценке только полностью решенных подзадач рекомендуется указывать результат на каждом тесте или баллы за подзадачу или вариант «первая ошибка» как на региональном и заключительном этапе (указывается номер первого не прошедшего теста и тип ошибки на нем).

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 определённым количеством баллов.

Система оценивания каждой задачи указывается в условии задачи. Если используются общие схемы оценивания в разных задачах, например, для каждой задачи указаны подзадачи и определены 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.

Тесты, приведённые в условии задачи в качестве примеров, оцениваются в 0 баллов.

Для школьного этапа в качестве основной рекомендуется потестовая система оценки. Исключения составляют задачи с ответами вида «Да/нет» (их не следует давать при потестовой системе оценки, т. к. тогда легко пишется решение, набирающее большое количество баллов) и т. п. Если используется потестовая система оценки и одним из вариантов вывода программы является «No solution» или его аналоги (вывод специального значения, например, числа 0 или -1 при отсутствии решения), то тесты, в которых ответ является таким, должны оцениваться не более, чем в 20% от полного балла. Допускается также специальное условие для оценивания, что решения, выдающие правильный ответ только на тестах вида «No solution» оцениваются в 0 баллов.

3.4.6. Использование тестирующей системы

Организаторы школьного или муниципального этапа могут установить и настроить собственный экземпляр тестирующей системы либо использовать тестирующую систему, доступную по модели «software as a service», например:

- Яндекс-контест https://contest.yandex.ru;
- Codeforces https://codeforces.com;
- Система ОЦ «Сириус» https://uts.sirius.online.

В случае если школьный этап для всего региона проводится по общим заданиям, рекомендуется использование общей тестирующей системы для всего региона. Муниципальный этап рекомендуется проводить с использованием общей тестирующей системы для всего региона.

3.4.7. Необходимое материально-техническое обеспечение

В дополнение к материально-техническому обеспечению, указанному в разделе 4, 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;
- редакторы электронных таблиц.

4. Необходимое материально-техническое обеспечение для выполнения олимпиадных заданий школьного и муниципального этапа олимпиады

4.1 Материально-техническое обеспечение при использовании заданий в бланковой форме

Задания тиражируются на листах бумаги формата А4 или А5, решения заданий записываются в тетрадях, на отдельных листах или специальных бланках. Для черновых записей участникам предоставляется бумага, черновики сдаются после окончания олимпиады, но не проверяются.

4.2. Материально-техническое обеспечение при компьютерной форме проведения этапа

Каждый участник должен быть обеспечен рабочим местом, оснащённым 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. Доступ в Интернет рекомендуется запретить, за исключением при необходимости доступа к серверу тестирующей системы.

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 дискомфорта другим

участникам олимпиады. На используемые клавиатуры и мыши могут быть наложены дополнительные требования.

Задания тиражируются на листах бумаги формата А4 или А5. Допускается предоставление доступа к электронным версиям заданий в интерфейсе тестирующей системы. В случае предоставления электронных версий заданий распечатанные задания могут не предоставляться.

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

Дополнительное материально-техническое обеспечение (программное обеспечение, компиляторы, среды разработки) при использовании на соответствующих этапах различных видов задач приведено в описании этих видов задач в разделе 3.

Соответствующая предметно-методическая комиссия заранее утверждает список программного обеспечения, который будет использоваться для проведения школьного и муниципального этапов и доводит эту информацию до сведения участников и организаторов олимпиады.

5. 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 флеш-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.

6. Критерии и методика оценивания выполненных олимпиадных заданий

Система и методика оценивания олимпиадных заданий должна позволять объективно выявить реальный уровень подготовки участников олимпиады.

Принципы формулирования критериев и методики оценки олимпиадных заданий для каждого типа заданий приведены в соответствующих пунктах раздела 3.

7. Использование учебной литературы и интернет-ресурсов при подготовке школьников к олимпиаде

При подготовке участников к школьному и муниципальному этапам олимпиады целесообразно использовать следующие нижеприведенные источники.

1. https://informatics.msk.ru – сайт дистанционной подготовки к олимпиадам по информатике
2. https://edu.sirius.online – Сириус курсы
3. https://neerc.ifmo.ru/school – архив материалов различных олимпиад по информатике для школьников
4. https://codeforces.com – сайт онлайн-соревнований по информатике для разного уровня сложности

Приложение 1

Примеры заданий

ПЯТИЗНАЧНОЕ ЧИСЛО

(5-6 классы, бланковая форма)

Условие.

В пятизначном числе не меньше трёх цифр, которые меньше 5, и не меньше трёх нечётных цифр. Найдите самое большое из таких чисел. Объясните, почему найденное вами число является наибольшим.

Решение.

На первое место числа поставим наибольшую из возможных цифр – 9. На второе место также можно поставить цифру 9, и ещё останется три цифры. При этом оставшиеся три цифры должны быть меньше 5, из них хотя бы одна должна быть нечётная (так как две нечётные цифры уже были записаны). Наибольшая цифра, которая меньше 5, – это 4, наибольшая нечётная цифра, которая меньше 5, – это 3. Значит, среди трёх оставшихся цифр можно использовать две цифры 4 и одну цифру 3. Чтобы число было наибольшим, необходимо сначала записать две цифры 4, потом одну цифру 3.

Ответ – 99443.

Критерии оценивания.

Правильный ответ с объяснением – 5 баллов.

Только ответ без объяснения – 4 балла.

Ответы 99344, 99434 (т. е. перестановка цифр из правильного ответа) – 2 балла.

Ответы 99333, 98433 – 2 балла.

Любое другое пятизначное число, в котором не меньше трёх цифр меньше 5 и не меньше трёх нечётных цифр (т. е. не выполнено только условие максимальности), – 1 балл.

Примечание.

Это задание можно проверять автоматически при помощи тестирующей системы.

КВИДДИЧ
(5-6 классы, бланковая форма)

Условие.

В вымышленной спортивной игре квиддич соревнуются две команды. Каждый гол, забитый в ворота противника, приносит команде 10 очков. Если же игрок одной из команд поймает специальный мяч – снитч, то эта команда получает дополнительные 150 очков, после чего игра заканчивается.

В финале очередного чемпионата Хогвартса по квиддичу встретились команды Когтеврана и Пуффендуя. На протяжении всего матча команды сражались на равных, разница в счёте никогда не превышала 10 очков (т. е. одного гола), и в конце матча лидировал Когтевран, но благодаря пойманному снитчу победил Пуффендуй. Также после окончания матча журналисты опросили всех игроков, забивших хотя бы один гол.

Алиса сказала, что забила только один гол – на 27-й минуте.

Боб забил один гол на 30-й минуте.

Виктория забила два гола – на 5-й и 21-й минутах.

Глория забила четыре гола на 10, 12, 34 и 53-й минутах.

Дональд забил два гола на 14-й и 42-й минутах.

Эдвард забил три гола на 15, 23 и 56-й минутах.

Выполните задания:

1. Укажите, с каким счётом закончилась игра (не забудьте, что снитч приносит 150 очков).

2. Для всех перечисленных игроков укажите, за какую команду они играли.

Решение.

Игроков будем обозначать первой буквой их имени (А, Б, В, Г, Д, Э). Упорядочим по возрастанию моменты времени, в которые были забиты голы в матче, с указанием того, кто забил эти голы. Пока неясно, кто за какую команду играл, поэтому обозначим команды 1 и 2. Для заполнения строк «Команда» и «Счёт» воспользуемся условием: «На протяжении всего матча команды сражались на равных, разница в счёте никогда не превышала 10 очков (т. е. одного гола)».

Минута	5	10	12	14	15	21	23	27	30	34	42	53	56
Кто забил	В	Г	Г	Д	Э	В	Э	А	Б	Г	Д	Г	Э
Команда	1	2	2	1	2	1	2	1	1	2	1	2	2
Счёт	10:0	10:10	10:20	20:20	20:30	30:30	30:40	40:40	50:40	50:50	60:50	60:60	60:70

Поскольку сказано, что в конце матча лидировал Когтевран, то команда 2 – это Когтевран, а команда 1 – Пуффендуй. Но поскольку снитч поймал Пуффендуй, то Пуффендуй выиграл со счётом 210 : 70.

За Когтевран (команда 2) играли Глория и Эдвард, за Пуффендуй (команда 1) играли Алиса, Боб, Виктория и Дональд.

Критерии оценивания.

Оценка за задание (максимум 5 баллов) складывается из суммы двух оценок – указание итога матча (максимум 2 балла) и указание того, за какие команды играли те или иные игроки (максимум 3 балла).

За правильно указанный итог матча ставится 2 балла. Если при подсчёте очков не учтён финальный снитч (указан счёт 60 : 70 в пользу Когтеврана), то ставится 1 балл.

За правильное указание того, в каких командах играли какие игроки, – 3 балла. Если команды полностью перепутаны местами (Глория и Эдвард указаны в Пуффендуе, остальные – в Когтевране), то ставится 2 балла. Если при восстановлении хронологии матча допущена одна ошибка – 1 балл.

ПЕРЕПРАВА

(5-6 классы, бланковая форма)

Условие.

К реке подошли волчица с тремя волчатами и лисица с тремя лисятами. Зверям необходимо переправиться на другой берег. У берега привязана лодка, которая вмещает только двух зверей. Ситуация осложняется тем, что волчица с лисицей не доверяют друг другу и не оставят своих детей в своё отсутствие с другой мамой ни на берегу, ни в лодке. Грести умеют только лисица и один из лисят. Как им переправиться? Постарайтесь составить как можно более короткий план переправы.

Решение.

Обозначим лисёнка, который умеет грести, как «лисёнок1». Возможный план перевозки:

1. Перевезти лисёнка1 и лисёнка
2. Перевезти лисёнка1
3. Перевезти лисёнка1 и лисёнка
4. Перевезти лисёнка1
5. Перевезти лисицу и лисёнка1
6. Перевезти лисицу
7. Перевезти лисицу и волчицу
8. Перевезти лисёнка1

9. Перевезти лисёнка1 и волчонка
10. Перевезти лисёнка1
11. Перевезти лисёнка1 и волчонка
12. Перевезти лисёнка1
13. Перевезти лисёнка1 и волчонка

Критерии оценивания.

Полностью правильное описание перевозки без лишних действий – 5 баллов.

При наличии не более 2 лишних действий – 4 балла.

При наличии не более 4 лишних действий – 3 балла.

Любой правильный алгоритм перевозки без учёта числа лишних действий – 2 балла.

Примечание.

Если формализовать условие этой задачи и строго описать формат записи плана переправы, возможна автоматическая проверка такого задания.

ВЗВЕШИВАНИЯ

(5-6 классы, бланковая форма)

Условие.

Есть шесть гирек, известно, что их массы равны 1, 2, 3, 4, 5 и 6 граммов, но размеры гирек одинаковые. На гирьках написаны цифры: 1, 2, 3, 4, 5, 6. Также есть чашечные весы. Эксперт знает, что на каждой гирьке верно записана её масса, но судья в этом сомневается. Как эксперт может убедить в этом судью? Какое минимальное количество взвешиваний ему необходимо для этого сделать?

Решение.

Задачу можно решить за два взвешивания.

Первым взвешиванием эксперт кладёт на одну чашу весов гирьки с цифрами 1, 2 и 3, на другую чашу весов – гирьку с цифрой 6. Весы останутся в равновесии. Такое возможно только при взвешивании самой тяжёлой гирьки (6) с тремя самыми лёгкими (1, 2, 3).

После этого взвешивания стало известно, что:

а) На гирьке массой 6 правильно обозначена её масса.

б) Гирьки, подписанные 1, 2, 3, имеют массу 1, 2 и 3, но, возможно, в другом порядке.

в) Две оставшиеся гирьки, подписанные 4 и 5, также имеют массу 4 и 5, но, возможно, в другом порядке.

При втором взвешивании эксперт на одну чашку весов положит гирьки с цифрами 1 и 6, а на другую – гирьки с цифрами 3 и 5. Поскольку $3 + 5 > 1 + 6$, то гирьки 3 и 5 перевесят.

Такое возможно только в том случае, если из двух групп (1, 2, 3 и 4, 5) взяли самые тяжёлые гирьки (3 и 5), а к гирьке 6 добавили самую лёгкую (1). Тем самым точно установлена масса гирек, подписанных 1, 3, 5, оставшаяся гирька из первой группы имеет массу 2, из второй группы – массу 4.

Критерии оценивания.

Правильное решение за два взвешивания – 5 баллов.

Правильное решение за три взвешивания – 3 балла.

Правильное решение за четыре взвешивания – 2 балла.

Правильное решение за любое число взвешиваний – 1 балл.

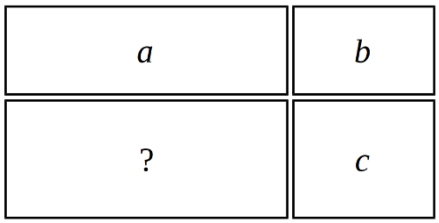
Неправильный алгоритм, но правильно указано взвешивание 1 + 2 + 3 = 6 (оно даёт наибольшую информацию о гирьках) – 1 балл.

ПЕРИМЕТР

(7-8 классы, компьютерная форма)

Условие.

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



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 «2*a*» 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 «Неправильный формат записи 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 классы, 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 местонахождения 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 озере 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 «Неправильный формат записи ответа».

При окончательной проверке проверяющая программа выполняет моделирование действий Гарри Поттера, определяя все возможные озёра, в которых может находиться рыбка после очередного хода, т.е. проверяется, действительно ли указанная последовательность действий Гарри Поттера позволяет всегда поймать рыбку, будем называть такие решения *корректными*.

Корректное решение, состоящее из 6 чисел, получает максимальный балл, другие корректные решения получают меньшее число баллов, в зависимости от длины ответа. Рекомендуется за любое корректное решение, независимо от его длины, давать 30–50% от максимального балла.

Также можно небольшим числом баллов оценивать решения, не являющиеся корректными, 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 классы, 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 «Неправильный формат записи 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 классы, 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 лёгкий груз.

Масса ювелирного изделия, которую нужно определить ювелиру, является целым числом от 1 до 25 граммов. 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 граммов. 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 100 баллов, из четырёх гирек – 50 баллов, из пяти гирек – 20 баллов.

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 «Неправильный формат записи 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 100 баллов, правильное решение из 4 гирек (например, 1 3 9 18) оценивается в 50 баллов, решение из 5 гирек (например, 1 2 4 8 16) оценивается в 20 баллов.

ДВА ПОДАРКА

(9-11 классы, 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 ($2 \le n \le 100000$, $2 \le x \le 10^9$).

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

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 классы, 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 подаётся одно число n ($1 \le n \le 100000$).

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 $kx \le n$, прибавляем к $d[kx]$ единицу. Чтобы найти ответ на задачу, нам нужно просто найти максимум в этом массиве.

РОДИТЕЛЬСКИЙ СОВЕТ

(9-11 классы, 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 ($N > 0$, $0 \le K \le N$), записанных 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 $2 \cdot 10^9$, будет оцениваться в 100 баллов.

Пример.

Ввод	Вывод
27 7	3

СЧАСТЛИВЫЕ БИЛЕТЫ

(9-11 классы, компьютерная форма)

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 зашёл 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. Всё, 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 её 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 чётким. 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 сложноподчинённых 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 тоже не рекомендуется использовать в условиях). Следует писать полностью «то есть», «так как».

7. Последний абзац условия должен резюмировать условие и ещё раз чётко 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 нём также можно ещё раз повторить, что требуется найти.

Особенности при подготовке условия в системе вёрстки 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

Выведите 2^n чисел – ОК

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. Б***о***льшие – числом.

Например,

У Васи было три поросёнка – ОК

У Васи было 3 поросёнка – неправильно

У Пети было три тысячи пятьсот двенадцать поросят – неправильно

У Пети было 3512 поросят – ОК

18. Порядковые числительные с параметром либо б***о***льшие 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 и т. п.).

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 задаёте ограничение сразу на несколько переменных, пишите их через запятую. В этом случае, если у вас подряд идёт 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 о том, что такой случай бывает.

28. Примеров не должно быть слишком много.

Выбор ограничений и написание решения.

29. По каждой задаче должно быть решение на Паскале, Python, C++ или Java, которые написаны естественным образом без неасимптотических оптимизаций (например, быстрого ввода) и укладываются в TL с двухкратным запасом. Рекомендуется использовать для написания эталонного решения язык Python, как наиболее медленный из всех распространённых языков.

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. Не рекомендуется использовать «мультитесты», то есть несколько тестовых наборов для одного запуска программы, так как описание мультитестов запутывает условие подробностями, не имеющими отношения к содержанию задачи.

36. Большие тесты необходимо сгенерировать, генератор тестов можно, например, писать на C++ с использованием библиотеки testlib.

37. Тесты должны быть корректными текстовыми файлами. Каждая строка, включая последнюю, должна завершаться переводом строки.

38. Тестирование может проводиться как под Windows, так и под Linux. Перевод строки под Windows задаётся двумя символами: 13 и 10 в этом порядке. Перевод строки под Linux задаётся 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”. Не используйте print для вывода более чем одной строки.

39. 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.

40. Данные во входном файле должны быть разбиты на строки в точности так, как описано в условии задачи. Лишних данных в тестах быть не должно.

41. Генератор тестов должен быть детерминированным. Он должен выдавать одни и те же тесты при повторных запусках.

42. 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.

43. 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.

44. Для избегания ошибок при подготовке тестов рекомендуется использовать валидаторы – специальные программы, проверяющие корректность тестов.

45. 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.

46. 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.

47. Для написания валидаторов можно применять библиотеку testlib.

Приложение 3

Рекомендуемые интернет-ресурсы 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#, – https://visualstudio.microsoft.com/ru/vs/ (необходимо использовать Community edition);

– Visual Studio Code – https://code.visualstudio.com/. Для работы данной среды разработки необходимо отдельно установить языки программирования (GNU C++, Python и т. д.) и после этого под логином участника олимпиады установить дополнительные расширения для поддержки языков программирования. Рекомендуются расширения «C/C++ Extension Pack», «Python Extension Pack»).

– Oracle Java – https://www.oracle.com/java/technologies/downloads;

– OpenJDK Java – https://openjdk.org/projects/jdk/20//;

– Python – https://www.python.org/;

– PascalABC.Net – http://pascalabc.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/.

– Libre Office: https://ru.libreoffice.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 этапов некоторых регионов

1. Москва https://olympiads.ru/moscow/
2. Санкт-Петербург http://neerc.ifmo.ru/school/spb/municipal.html
3. Московская область https://mosregolymp.mipt.ru/
4. Подборка заданий из разных регионов https://olimpiada.ru/activity/73/tasks
5. Олимпиады, проводимые ОЦ «Сириус» https://siriusolymp.ru/