Краткие методические указания
по решению задач заключительного этапа
Олимпиады школьников СПбГУ по информатике.

7-9 класс

Вариант олимпиадного задания состоит из четырех задач. Оценивание задач происходит по первичным баллам. Максимальное количество баллов, которые могут быть начислены за правильные ответы по каждой задаче, представлены в таблице:

Задача №	***Максимальное количество баллов (7-9 класс)***
1	3
2	7
3	7
4	8

Подсчет первичной оценки осуществляется путем суммирования первичных баллов, выставленных за ответы на каждый из вопросов. Максимальная сумма первичных баллов по всем правильно решенным задачам равна **25**. Перевод первичных баллов в итоговые осуществляется умножением первичных баллов на коэффициент 4. Максимальная сумма итоговых баллов по всем правильно выполненным задачам равна **100**.

Задание №1.

> Есть 20 монет. Среди них имеется 1 фальшивая. Однако неизвестно легче она или тяжелее. Определить какое наименьшее количество взвешиваний на обычных весах (типа «коромысло») нужно совершить, чтобы однозначно определить фальшивую монету. Ответ обоснуйте.

Цель задачи: проверить логическое мышление (задача творческого характера).

Схема решения:

При решении подобного рода задач необходимо описать не только последовательность взвешиваний но и возможные ситуации, возникающие при взвешивании.

1. Необходимо разбить монеты на несколько кучек.
2. Попарным взвешиванием выяснить какая кучка отличается от других.
3. Повторить п.1 и п.2 до тех пор пока не останется 1 или 2 монеты.
4. Если ранее не было сделано выводов о весе монеты то, сравнительным взвешиванием с другими монетами, выяснить какая является фальшивой.

Замечание 1: первоначальное разбиение на 2 кучки не всегда является оптимальным.

Замечание 2: подходы к решению бывают различными. Количество монет в кучках может быть разным. Самым важным в этой задаче является объяснение последовательности взвешиваний и описание логических выводов, непосредственно следующих из каждого взвешивания.

Задание №2.

> На корабле есть 3 зеленые лампы, 3 синие лампы и 4 красные. Сколько всевозможных комбинаций сигналов может подать корабль при условии, что никакие 2 одинаковые лампочки не стоят рядом.

Цель задачи: проверить логическое мышление (задача творческого характера).

Схема решения:

Возможны 3 варианта решения задачи, разные по степени творческого подхода.

Решение №1 (элементарное)

> При оформлении этого решения количество ламп в сигнале фиксировано и равно количеству всех цветных ламп.

Решение №2 (стандартное)

При оформлении решения последовательно учитывается то что сигнал может состоять из 1 лампы, из двух ламп, из трех и т.д. ламп. В этом подходе решение, описанное ранее, является частным случаем.

Решение №3 (комплексное)

При оформлении этого решения учитывается то что лампа может гореть, а может не гореть и тогда важен не только порядок и количество ламп в сигнале, но и положение лампы в общем ряду.

Замечание: Основная нагрузка в решении сводится к наиболее полному описанию возможных вариантов сигналов и вычислению их количества.

Задание №3.

Решить систему уравнений относительно x,y:

$$\begin{cases} 21_x \cdot 30_y = 420_x \\ 10_y \cdot 10_x = 120_y \end{cases}$$

Цель задачи: проверить знание систем счисления и умение проводить действия с ними.

Схема решения:

Для решения подобного рода задач необходимо:

1. преобразовать каждое число в уравнение (из системы счисления X, У перейти в десятичную систему счисления);
2. составить систему уравнений;
3. решить составленную систему уравнений;
4. получить либо единственное решение, либо множество решений, либо сделать вывод об отсутствии каких-либо решений в целых числах. При этом надо понимать, что основанием (т.е. решением системы) не может быть целое число, меньшее максимальных значений цифр в первоначальном представлении системы.

Задание №4.

Задана строка из **n** слов. Необходимо написать **алгоритм** программы, позволяющий развернуть каждое слово в строке задом наперед, при этом порядок следования слов сохраняется.

Цель задачи: проверить начальные знания программирования.

Схема решения:

Для решения подобного рода задач необходимо:

1. описать процесс ввода строки (тип переменной);
2. описать процесс выделения слова (считывание букв от пробела до пробела, ведение счетчика считанных букв);
3. описать процесс разворачивания слова (обратный цикл с использованием найденного количества букв в слове и записью нового слова в отдельную переменную);
4. зацикливание процесса т.е. переход к следующему слову (повтор п.2) до тех пор пока строка не кончится;
5. выведение на экран результата действий.

Замечание: основная нагрузка в решении сводится к наиболее полному описанию всех действий (в случае написания программ – к наиболее полным комментариям к программе и правильной работе программы).

Краткие методические указания
по решению задач заключительного этапа
Олимпиады школьников СПбГУ по информатике.

10-11 класс

Вариант олимпиадного задания состоит из четырех задач. Оценивание задач происходит по первичным баллам. Максимальное количество баллов, которые могут быть начислены за правильные ответы по каждой задаче, представлены в таблице:

Задача №	***Максимальное количество баллов (10-11 класс)***
1	2
2	5
3	8
4	10

Подсчет первичной оценки осуществляется путем суммирования первичных баллов, выставленных за ответы на каждый из вопросов. Максимальная сумма первичных баллов по всем правильно решенным задачам равна **25**. Перевод первичных баллов в итоговые осуществляется умножением первичных баллов на коэффициент 4. Максимальная сумма итоговых баллов по всем правильно выполненным задачам равна **100**.

Задание №1.

Решить систему уравнений относительно x,y:

$$\begin{cases} 21_x \cdot 30_y = 420_x \\ 10_y \cdot 10_x = 120_y \end{cases}$$

Цель задачи: проверить знание систем счисления и умение проводить действия с ними.

Схема решения:

Для решения подобного рода задач необходимо:

5. преобразовать каждое число в уравнение (из системы счисления X, У перейти в десятичную систему счисления);
6. составить систему уравнений;
7. решить составленную систему уравнений;
8. получить либо единственное решение, либо множество решений, либо сделать вывод об отсутствии каких-либо решений в целых числах. При этом надо понимать, что основанием (т.е. решением системы) не может быть целое число, меньшее максимальных значений цифр в первоначальном представлении системы.

Задание №2.

В файле **in.txt** в одну строчку написано большое натуральное число (более ста знаков, неспособное поместиться в память, выделяемую для целого числа в большинстве реализаций языков программирования) и через запятую второе число - трехзначное. Требуется написать программу поиска остатка от деления первого числа на второе. Результат выполнения программы вывести в файл **out.txt**.

Пример ввода (сокращенный для удобства): 654684634354684646231312 9, 345

Вывод: 344.

Цель задачи: проверка знания и/или умения производить "длинную арифметику".
Схема решения:
Для решения подобного рода задач необходимо:

1. Считывание делимого и делителя;
   Замечание 1: считывание делимого происходит в текстовую переменную (либо в массив, что не всегда является оптимальным); при считывании делимого и делителя необходимо учесть, что их разделяют запятая и пробел.
   Замечание 2: по условию задачи исходные данные вводятся из файла (или с клавиатуры, в случае создания пользовательского интерфейса). При этом указан размер делителя. В каждом варианте олимпиадных задач размер делителя разный. Наиболее универсальным является программный код который учитывает ситуацию когда размера делителя определяется после считывания входных данных.
2. Отделение от делимого части (с левого края), состоящей из ***k*** знаков, и перевод этой части в числовую переменную. Делитель (если это не сделано ранее) так же переводится из текстовой в числовую переменную.
   Замечание 1: после отделения ***k*** знаков, можно удалить их из текстовой переменной тем самым сократив ее длину. Построенный цикл считывания основан на постепенном уменьшении длинны строки содержащей делимое.
   Замечание 2: при отделении части делимого иногда можно учитывать, что она должна быть больше чем делитель. Такая проверка может сократить количество шагов.
3. Проводится целочисленное деление и получение остатка от такого деления.
4. Полученный остаток умножается на 10 и к полученному числу прибавляется цифра стоящая на ***k+1*** в первоначальной записи делимого. Дальнейшие действия происходят с полученным числом.
   Замечание: если в п.2 алгоритм учитывает замечание №1 то в п.4 добавляется первый (слева) знак из делимого т.к. после удаления ***k*** символов из строки, следующий (***k+1***) символ перемещается на первую позицию.
5. Последовательное повторение пунктов №3, 4, 5 до тех пор пока делимое (в п.4.) больше чем делитель или не исчерпаются все знаки в делимом.
6. Полученное значение выводится на экран.

Замечание. В п.3. может быть применен альтернативный подход: Пусть имеется делимое ***A*** и делитель ***B***; пусть ***C*** является целой частью при делении ***A*** на ***B***, а ***D*** является остатком от деления т.е. ***A=C*B+D***. Возможен следующий подход: ***A*** делится на ***B***; у полученного числа выделяется целая часть ***C***; остаток ***D*** находится по формуле: ***D=A - C*B***. Дальнейшие действия аналогичны представленным в алгоритме.

Задание №3.
Цель задачи: проверка написания процедур и функций, а так же умения работать с массивами и проверка (в некоторых задачах: проверка умения осуществлять сортировки и выявлять взаимосвязи между элементами).

Дан двумерный массив размером NxN, заполненный символами A-Z. Написать программу, осуществляющую поиск в данном массиве строк размером <= N, заданных в файле и вывод координат первой буквы каждого слова и указателя направления слова. Строки могут располагаться горизонтально, вертикально и диагонально, и при этом быть написаны в любую сторону. Указатель направления – это направление, в котором читается строка в массиве (С, Ю, З, В, СЗ, ЮВ и т.д.). Если строка присутствует в массиве несколько раз, то необходимо вывести все ее вхождения в массив. Если же строка не встречается в массиве, то явно указать

это.

Пример:

Вход:	Выход:
4 Q N O S E A T N Z N O O A B O D DO NOT EAT ANTS	DO (4, 4) З (4, 4) С (4, 4) СЗ NOT Не найдено EAT (2, 1) В ANTS (4, 1) СВ

Схема решения:

Для решения подобного рода задач необходимо:

1. Осуществить ввод исходных данных, выделить массив для поиска и слова которые необходимо найти.
2. Взять первую букву слова (которое необходимо найти) и найти ее в заданном массиве (найти все ее положения).
3. Последовательной проверкой по разным направлениям проверить совпадение других букв массива соответствующим буквам искомого слова. В случае если поиск слова завершился удачно необходимо вывести направление слова. В случае не удачи – продолжить поиск по другим направлениям. В случае исчерпания направлений и не удачи в поиске слова – выдать "не найдено".

Замечание 1: необходимо учесть, что один раз выбрав направление, последующую проверку, на соответствие букв искомому слову, следует вести строго в этом направлении.

Замечание 2: проверку по направлениям можно объединить в одну процедуру или функцию. Подобное объединение следует считать наиболее оптимальным (и удобным для описания с помощью комментариев).

Замечание 3: проверку можно упростить если, сравнить длину искомого слова с возможной длинной слова в выбранном направлении (предотвращает выход за пределы массива).

4. Повторить п.2 для нового слова. По исчерпании слов для поиска – завершить работу программы.

Замечание: возможен иной подход к решению задачи. Для каждой буквы (ячейки) массива составляется множество возможных значений (по всем возможным направлениям) с запоминанием направления для каждого слова; далее производится сортировка массива; окончательный поиск для искомого слова происходит по отсортированному массиву с выведением значения для направления каждого слова. Подобный подход не всегда бывает оптимальным т.к. требует создания дополнительного массива большой размерности. Однако, бывают задачи когда это оправдано т.к. составленный словарь значительно упрощает поиск искомого слова.

Задание №4.

Цель задачи: проверка знания принципов раскраски и обхода лабиринта, а так же принципов написания процедур и функций.

Задача №4.

В системе подземных туннелей началось подтопление. Вода двигается во все возможные направления с одинаковой скоростью. В некоторых местах туннелей установлены датчики подтопления. Требуется, имея схематический план туннелей в файле in.txt, в out.txt вывести номер датчика, который сработает первым.

Формат ввода: двумерный массив из символов (максимум 15x15), схематически отображающий план туннелей, в котором символы 'x' обозначают стены, нули – свободное пространство. ‘T’ обозначает источник воды, числа от 1 до n – номера датчиков. Длины коридоров прямо пропорциональны числу нулей, в их представлении.

Пример:

Ввод

```
x x x x x x x x x x x x x x x x x
x x x x 5 x x x x x x x x x x x x
x x x x 0 x x x x x x x x 1 x x x
x x x x 0 x x x x x x x x 0 x x x
x x x x 0 x x x x x x x x 0 x x x
x x x x 0 x x 0 0 0 0 0 0 x x x x
x x x x 0 x x 0 x x x x x 0 x x x
x 4 0 0 0 0 x 0 x x x x x 0 x x x
x x x x x 0 x T x x x x x 0 x x x
x x x x x 0 0 0 0 0 0 0 0 x x x x
x x x x x x x x x x 0 x x 2 x x x
x x x x x x x x x x 0 x x x x x x
x x x x x x x x x x 0 0 3 x x x x
x x x x x x x x x x x x x x x x x
```

Вывод: 2

Схема решения:

Для решения подобного рода задач необходимо:

1. Считать исходные данные в массив.
2. Найти координаты начальной точки "T" и обнулить счетчик пройденного пути.
3. Проверка соседних ячеек массива на наличие в них "0" и, в случае его обнаружения, перейти в эту позицию. Текущее значение счетчика пути увеличивается на 1. Действие пункта повторяется до тех пор пока не будет найдена цифра (значение отличное от "0" и "x").

   Замечание 1: проверка осуществляется вправо, влево, вверх и вниз от текущей позиции. Проверка элемента по диагонали приведет к грубой ошибке (сокращение пути, не рассмотрение некоторых решений).

   Замечание 2: проверку по направлениям можно свести в единую процедуру, входными параметрами которой будет направление обхода т.е. создание универсально процедуры.
4. В случае обнаружения в соседней ячейке цифры (не "0" и не "x"), вычисляется длина пройденного пути.
5. Вычисляется минимальная длина среди всех контрольных точек (обозначенных цифрами) и выводится номер, соответствующий минимальной длине.

Возможен альтернативный подход к решению задачи (раскраска):

1. Рассматриваются два массива. В одном хранится лабиринт, в другом, временном (заполняется "-1"), формируется план обхода лабиринта.
2. Находятся координаты контрольных точек.
3. Находится начало пути, точка "Т" (текущее значение счетчика =0).
4. Проверяются соседние (справа, слева, сверху, снизу от текущей точки) ячейки.

   Если в них содержится "0", то во временном массиве в соответствующей ячейке ставится значение счетчика (увеличивается на 1 от текущего для предыдущей точки).

   Если в них содержится число, и координаты этой точки не совпадают с координатами контрольных точек (т.е. наткнулись на пересекающийся путь) то если число больше текущего значения счетчика (т.е. предыдущий обход длиннее) то заменяем на текущее, если меньше (т.е. другой обход короче) то оставляем без изменения, прекращаем обход и возвращаемся к другим направлениям.
5. Для каждой контрольной точки во временном массиве исследуются соседние точки (справа, слева, сверху, снизу).

   Если среди соседних ячеек найдено только одно положительное число то оно увеличивается на 1 и записывается во временный массив на место контрольной точки.

   Если найдено несколько положительных цифр (т.е. существуют несколько подходов к точке) то среди них выбирается минимальное значение, увеличивается на 1 и записывается во временный массив на место контрольной точки.
6. Выбирается минимально значение, стоящее на местах контрольных точек во временном массиве, и выводится значение для этой точки из основного массива с лабиринтом.

Существуют и другие способы раскраски.

Оформление задач по программированию.

Каждая работа содержит требования к оформлению. За не выполнение этих требований максимальное количество первичных баллов может быть уменьшено по каждому из нижеследующих пунктов:

- за написание программ на языке отличном от C, C++ или Pascal: на 2 балла;
- за отсутствие операций с файлами входных и выходных данных или за отсутствие понятного пользователю интерфейса ввода-вывода исходных данных: на 1 балл;
- за отсутствие комментариев, облегчающих понимание кода программы:

Задача	***Количество штрафных баллов (10-11 класс)***
1	1*
2	2
3	3
4	4

* в случае если в качестве решения задачи предлагается программный код.

ВЫПИСКА
ИЗ ПРОТОКОЛА № 16
ЗАСЕДАНИЯ МЕТОДИЧЕСКОЙ КОМИССИИ ПО ПРОВЕДЕНИЮ ОЛИМПИАДЫ ШКОЛЬНИКОВ СПбГУ ПО ИНФОРМАТИКЕ

от 30 марта 2015 года

ПРИСУТСТВОВАЛИ:

Должность	Ф.И.О.	Должность на факультете, учёная степень, учёное звание
Зам.председателя	Андрианов Сергей Николаевич	Заведующий кафедрой компьютерного моделирования и многопроцессорных систем СПбГУ, профессор, д.ф.-м.н.
Зам.председателя	Дегтярев Александр Борисович	Профессор кафедры компьютерного моделирования и многопроцессорных систем СПбГУ, профессор, д.т.н.
Член комиссии	Моисеев Игорь Анатольевич	Доцент кафедры компьютерных технологий и систем СПбГУ, к.ф-м.н.

Повестка дня:

1. Установление критериев проверки заданий заключительного этапа Олимпиады школьников СПбГУ по информатике.

СЛУШАЛИ:

1. Проект Критериев проверки заданий заключительного этапа Олимпиады школьников СПбГУ по информатике.

ПОСТАНОВИЛИ:

1. Принять критерии проверки заданий заключительного этапа Олимпиады школьников СПбГУ по информатике в следующей редакции:

Критерии оценки заданий отборочного этапа
олимпиады по информатике в СПбГУ 2014-2015 уч. г.

Вариант олимпиадного задания состоит из четырех задач. Оценивание задач происходит по первичным баллам. Максимальное количество баллов, которые могут быть начислены за правильные ответы по каждой задаче, представлены в таблице:

Задача №	***Максимальное количество баллов (10-11 класс)***	***Максимальное количество баллов (7-9 класс)***
1	2	3
2	5	7
3	8	7
4	10	8

Максимальная сумма: 25 баллов.

По усмотрению жюри, за оригинальность приведенного решения (по каждой задаче), может быть начислен 1 бонусный балл. При этом, общая сумма первичных баллов по всем задачам не может превышать максимально возможной (25 баллов).

За не выполнение требований к оформлению задач, максимальное количество первичных баллов может быть уменьшено по каждому из нижеследующих пунктов:

- за написание программ на языке отличном от C, C++ или Pascal: на 2 балла;
- за отсутствие операций с файлами входных и выходных данных или за отсутствие понятного пользователю интерфейса ввода-вывода исходных данных: на 1 балл;
- за отсутствие комментариев, облегчающих понимание кода программы:

Задача	***Количество штрафных баллов (10-11 класс)***	***Количество штрафных баллов (7-9 класс)***
1	1*	-
2	2	-
3	3	2*
4	4	3*

* в случае если в качестве решения задачи предлагается программный код.

Подсчет первичной оценки осуществляется путем суммирования первичных баллов, выставленных за ответы на каждый из вопросов. Максимальная сумма первичных баллов по всем правильно решенным задачам равна **25**. Перевод первичных баллов в итоговые осуществляется умножением первичных баллов на коэффициент 4. Максимальная сумма итоговых баллов по всем правильно выполненным задачам равна **100**.

При проверке работы учитывается следующее (10-11 класс):

Задача 1. По условию задачи необходимо найти основание системы счисления при котором заданная система уравнений будет иметь решение, либо доказать что она не имеет решение.

0 баллов выставляется если решение не верно, либо к его решению не приступали.

1 балл выставляется в случае если ответ не верен полностью или частично т.к. решение содержит вычислительные ошибки, при этом более 50% решения является правильным.

2 балла выставляется за правильно решенную задачу.

Задача 2. По условию задачи необходимо составить программу на одном из языков программирования (Pascal или C/C++). При проверке задачи выставляется:

0 баллов если к решению задачи не приступили, либо программный код не содержит алгоритмических действий или предложенные действия представляют не более 10% от решения;

1 балл если программный код не учитывает условий задачи, либо программный код содержит не более 20-25% от решения;

2 балла если сделана попытка осуществления или объяснения хода решения, либо составленная программа имеет существенные ошибки и/или недочеты, либо не учтены требования задачи;

3 балла в случае если составленная программа имеет ошибки и/или недочеты, либо отсутствует вывод результата с учетом требования задачи;

4 балла в случае если составленная программа имеет не существенные ошибки и/или недочеты;

5 баллов выставляется за правильно решенную задачу.

Задача 3. По условию задачи необходимо составить программу на одном из языков программирования (Pascal или C/C++). При проверке задачи выставляется:

0 баллов если к решению задачи не приступили;

1 балл если программный код не содержит алгоритмических действий, либо предложенные действия представляют не более 10% от решения;

2 балл если программный код не учитывает условий задачи, либо программный код содержит не более 20-25% от планируемого решения, либо сделана попытка осуществления или объяснения хода решения, содержащее существенную ошибку;

3 балла если сделана попытка осуществления или объяснения хода решения, не содержащее существенных ошибок;

4-5 баллов в случае если составленная программа имеет существенную ошибку и/или недочет, либо отсутствует вывод результата с учетом требования задачи;

6-7 баллов в случае если составленная программа имеет не существенные ошибки и/или недочеты;

8 баллов выставляется за правильно решенную задачу.

Задача 4. По условию задачи необходимо составить программу на одном из языков программирования (Pascal или C/C++). При проверке задачи выставляется:

0 баллов если участник олимпиады к решению задачи не приступал;

1 балл если участник олимпиады приступил к решению задачи, однако ход решения полностью неверный, либо предложенные действия представляют не более 10% от решения;

2-3 балла если программный код не учитывает условий задачи и/или по окончании программы ответ принципиально отличается от условий задачи;

4-5 балла в случае если алгоритм решения задачи, предложенный участником олимпиады, является правильным, но решение содержит достаточно серьезные ошибки, влияющие на решение задачи;

6-7 баллов в случае если участник олимпиады в целом задачу решил правильно, но есть незначительные ошибки и/или не учтены ряд условий, влияющие на окончательный результат;

8-9 баллов если участник олимпиады в целом решил задачу правильно, но не учтены некоторые условия, влияющие на окончательный результат;

10 баллов выставляется за правильно решенную задачу с учетом всех возможных условий, влияющих на окончательный результат.

При проверке работы учитывается следующее (7-9 класс):

Задача 1. По условию задачи необходимо описать процесс поиска фальшивой монеты с помощью весов. При проверке задачи выставляется:

0 баллов выставляется если решение не верно, либо к его решению не приступали.

1 балл выставляется в случае если приведенное решение не верно полностью или частично, либо решение не соответствует существенным условиям задачи;

2 балл выставляется в случае если приведенное решение не верно полностью или частично;

3 балла выставляется за правильно решенную задачу.

Задача 2. По условию задачи необходимо описать комбинации, составляемые из различного набора сигнальных ламп. При проверке задачи выставляется:

0 баллов выставляется если к решению не приступали.

1 балла выставляется в случае если в качестве ответа приведено число без объяснения процесса его получения;

2 балла выставляется в случае если приведенное обоснование содержит не верные логические выводы;

3 балла выставляется в случае если приведенное обоснование не является логически завершенным и/или приведенное решение не соответствует существенным условиям задачи;

4 балла выставляется в случае если приведенное решение является логически завершенным, но не соответствует существенным условиям задачи;

5 баллов выставляется в случае если приведенное решение является логически завершенным, однако не учитывает всех возможных ситуаций;

6 баллов выставляется в случае если приведенное решение содержит описание всех возможных ситуаций, но содержит не существенные ошибки, влияющие на результат;

7 баллов выставляется за правильно решенную задачу.

Задача 3. По условию задачи необходимо найти основание системы счисления при котором заданная система уравнений будет иметь решение, либо доказать что она не имеет решение. При проверке задачи выставляется:

0 баллов выставляется если к решению не приступали.

1 балл выставляется если к решению задачи приступили, но полученный ответ является не верным, либо если указан ответ без описания его получения;

3 балла выставляется в случае если ответ не дан, но сделана верная попытка решения задачи и она содержит не более 50% от планируемого решения;

5 баллов выставляется в случае если ответ не верен полностью или частично т.к. решение содержит вычислительные ошибки, при этом более 50% решения является правильным.

7 баллов выставляется за правильно решенную задачу.

Задача 4. По условию задачи необходимо написать алгоритм. (Если в качестве ответа предлагается программный код то его оценивание осуществляется аналогично задаче №3 для 10-11 классов и с учетом штрафных баллов.) При проверке задачи выставляется:

0 баллов выставляется если к решению не приступали.

1 балл выставляется если к решению задачи приступили, но приведенное решение является не верным;

2 балла выставляется если приведено обобщенное решение, не содержащее детального рассмотрения ключевых моментов решения;

4 балла выставляется в случае если приведенное решение не верно частично, либо решение не соответствует существенным условиям задачи;

6 баллов выставляется в случае если решение верно, однако содержит ряд не существенных ошибок, либо не учитывает всего множества возможных ситуаций;

8 баллов выставляется за правильно решенную задачу.

Зам.председателя методической комиссии Олимпиады школьников СПбГУ по информатике — Дегтярев А.Б.