

**Ответы и краткие методические рекомендации по
проверке заданий муниципального этапа Всероссийской
олимпиады школьников по информатике и ИКТ
7-8 класс**

**Общие методические рекомендации по проверке решений
задач**

При проверке решений участников необходимо учитывать следующее.

1. Участникам муниципального этапа Всероссийской олимпиады школьников на каждом туре предлагается комплект из четырех задач, соответствующий классу обучения.

2. Задания 1-3 для 7-8 классов предполагают полное развернутое письменное решение, которое сдается участниками олимпиады на бумаге. Для проверки решений участников Региональная предметно-методическая комиссия по информатике подготовила для муниципального жюри один из вариантов решения, а также общие методические рекомендации по оцениванию работ учащихся.

3. Задание 4 для 7-8 класса предполагает решение в виде компьютерной программы, представленной на проверку в виде файла с исходным кодом на одном из разрешенных языков программирования. Методику проверки решения задания 4 см. в указаниях по проверке задания 4.

Задание 1. Число (40 баллов)

Тема: позиционные системы счисления.

Критерии оценивания.

Полное обоснованное решение и правильный ответ – 40 баллов.

Правильный ответ и обоснованы 2 цифры из трех – 30 баллов.

Правильный ответ и обоснована 1 цифра из трех – 20 баллов.

Правильный ответ без обоснования – 10 баллов.

В целом неправильный ответ, но указана хотя бы одна правильная цифра – 5 баллов.

Ответ: 15A

Решение.

Запишем исходное число в виде XYZ , результат умножения на 11 – ZXY и представим процесс умножения в столбик на 11:

$$\begin{array}{r} X \quad Y \quad Z \\ * \quad \quad \quad (11) \\ \hline Z \quad X \quad Y \end{array}$$

Старший разряд результата Z получается при умножении X на 11. При этом результат остается трехразрядным – переноса в 4-й разряд не происходит. Это значит, что как минимум (без учета переноса) $X * 11 < 16$. Поэтому $X=1$, а также $11 \leq Z \leq 15$, т.к. это цифра 16-ой системы счисления.

Перепишем умножение, подставив вместо X единицу:

$$\begin{array}{r} 1 \quad Y \quad Z \\ * \quad \quad \quad (11) \\ \hline Z \quad 1 \quad Y \end{array}$$

Старший разряд результата $Z = 1 * 11 + P$, где P – это возможный перенос из младшего разряда. Исходя из условия $11 \leq Z \leq 15$, проверяем возможные значения Z .

Пусть $Z = 15$. Переписываем умножение:

$$\begin{array}{r} 1 \quad Y \quad (15) \\ * \quad \quad \quad (11) \\ \hline (15) \quad 1 \quad Y \end{array}$$

Вычисляем Y : $15 * 11 = 165$.

Корректируем 165: $165 : 16 = 10$ (остаток 5). Поэтому $Y = 5$ и 10 – перенос. Для проверки выполняем умножение в 16-й системе счисления:

$$\begin{array}{r} 1 \quad 5 \quad (15) \\ * \quad \quad \quad (11) \\ \hline (11) \quad (55) \quad (165) \end{array}$$

Корректируем: $165 : 16 = 10$ перенос и остаток 5 – это цифра в разряде:

$$\begin{array}{r} 1 \quad 5 \quad (15) \\ * \quad \quad \quad (11) \\ \hline (11) \quad (55) \quad (165) \\ \hline (11) \quad (65) \quad 5 \end{array}$$

Корректируем 65: $65 : 16 = 4$ (остаток 1)

$$\begin{array}{r}
 \\
 \\
 \\
 \hline
 * \\
 \\
 \hline
 \\
 \\
 \hline
 \\

 \end{array}$$

Старший разряд результата $Z=15$, т.е. F. Противоречия с исходным предположением нет.

Т.о., исходное число – это 15F, результирующее – F15, что соответствует условию.

Пусть $Z = 14$. Переписываем умножение:

$$\begin{array}{r}
 \\
 \\
 \hline
 * \\
 \\
 \hline

 \end{array}$$

Вычисляем Y: $14 * 11 = 154$.

Корректируем 154: $154 : 16 = 9$ (остаток 10). Поэтому $Y = 10$ и 9 – перенос. Для проверки выполняем умножение в 16-й системе счисления:

$$\begin{array}{r}
 \\
 \\
 \hline
 * \\
 \\
 \hline

 \end{array}$$

Корректируем: $154 : 16 = 9$ перенос и остаток 10 – это цифра в разряде:

$$\begin{array}{r}
 \\
 \\
 \hline
 * \\
 \\
 \hline
 \\

 \end{array}$$

Корректируем 119: $119 : 16 = 7$ (остаток 7)

$$\begin{array}{r}
 \\
 \\
 \hline
 * \\
 \\
 \hline
 \\
 \\

 \end{array}$$

Старший разряд 18 корректируем: $18 : 16 = 1$ и 2 остаток. Происходит перенос и результат перестает быть 4-х разрядным. Кроме того, разряд №1 результата $Y=7$, а не 1. Получили противоречия.

По аналогии проверяются остальные значения Z.

Ответ: 15A.

Задание 2. Магический символ Луногории (60 баллов)

Тема: математическая логика.

Критерии оценивания. Правильный и обоснованный вариант ответа оценивается в 60 баллов. При оценивании решения руководствоваться следующими соображениями:

1. При наличии полностью правильного ответа:

- a. В случае, когда рассуждения, поясняющие сделанные выводы, полностью отсутствуют, то начислять 10 баллов.
- b. Если приведены рассуждения и сделаны правильные выводы только из 1 утверждения, то начислять 30 баллов.
- c. Если приведены рассуждения и сделаны правильные выводы из первого и второго утверждений, то начислять 40 баллов.
- d. Если приведены рассуждения и сделаны правильные выводы из всех тех утверждений, то начислять 60 баллов.

Решение.

1. Тот факт, что старший брат вначале утверждает, что уверен в том, что младший брат не знает как выглядит магический символ, означает только то, что этот символ не имеет форму квадрата или треугольника. Так как квадрат может быть желтым, а треугольник голубым, учитывая, что эти цвета не повторяются, то зная их, младший брат знал бы как выглядит магический символ и старший брат не мог бы утверждать, что уверен, что младший брат не знает, как выглядит магический символ. Следовательно, форма либо круг, либо пятиугольник.

2. Круг может быть синим или малиновым, а пятиугольник – малиновым, красным, или зелёным. Так как после слов старшего брата младший определил форму символа - он не мог быть малиновым, так как малиновый и круг, и пятиугольник.

3. После слов младшего брата старший брат определил, как выглядит магический символ, т.е. цвет символа (форму ему сообщил отец), если бы это был пятиугольник, то учитывая что, пятиугольников не малинового цвета

два: зелёный и красный старший брат не мог бы точно определить цвет магического символа, таким образом, делаем вывод, что магический символ это не малиновый круг.

Следовательно, это круг синий.

Ответ: Магический символ - синий круг.

Задание 3. Игра «Угадай число» (100 баллов)

Тема: алгоритмизация.

Идея решения.

Моделируем описанный процесс.

Указания к проверке.

Представленное участником решение проверяется вручную. Допускается любое правильное решение, использующее цикл. Однако, если предложен частично верный алгоритм, в котором не применяется цикл, то считать это частным случаем и оценивать из 80 баллов. В решениях, представляющих собой программы на языке программирования, наличие синтаксических ошибок, не искажающих замысел автора программы, не влияет на начисляемые баллы.

Критерии оценивания.

Решение должно содержать описание переменных и алгоритм (или программу).

1. Описание переменных (максимальный балл - 30):

- 30 баллов – описаны все использованных переменных, причем для каждой указан тип и назначение;

1. 25 баллов – имеется неполное описание всех переменные, в котором не менее двух переменные описаны полностью, но имеются переменные с неполным описанием (указан только тип, либо только назначение);

2. 20 баллов – полностью отсутствует описание 1 переменной, а для остальных дано полное описание;
3. 15 баллов – для всех переменных имеется только частичное описание, т.е. ни одна переменная не описана полностью;
4. 10 баллов – полностью отсутствует описание одной переменной, для части переменных имеется частичное описание, для части – полное;
5. 5 баллов – имеется описание, не подходящее под указанные выше критерии, в котором упоминаются не менее 50% переменных.

2. Алгоритм в виде блок-схемы, программы или описания на естественном языке (максимальный балл - 70). Оцениваются 2 составляющие алгоритма: организация повторяющихся действий, расчет и вывод количества попыток (максимум по 35 баллов за каждый пункт). Если для расчета используется другая формула, которая дает правильный результат, то за это баллы не снижаются. Если при правильном алгоритме не выведен ответ, то снимается 20 баллов.

Решение.

Описание переменных:

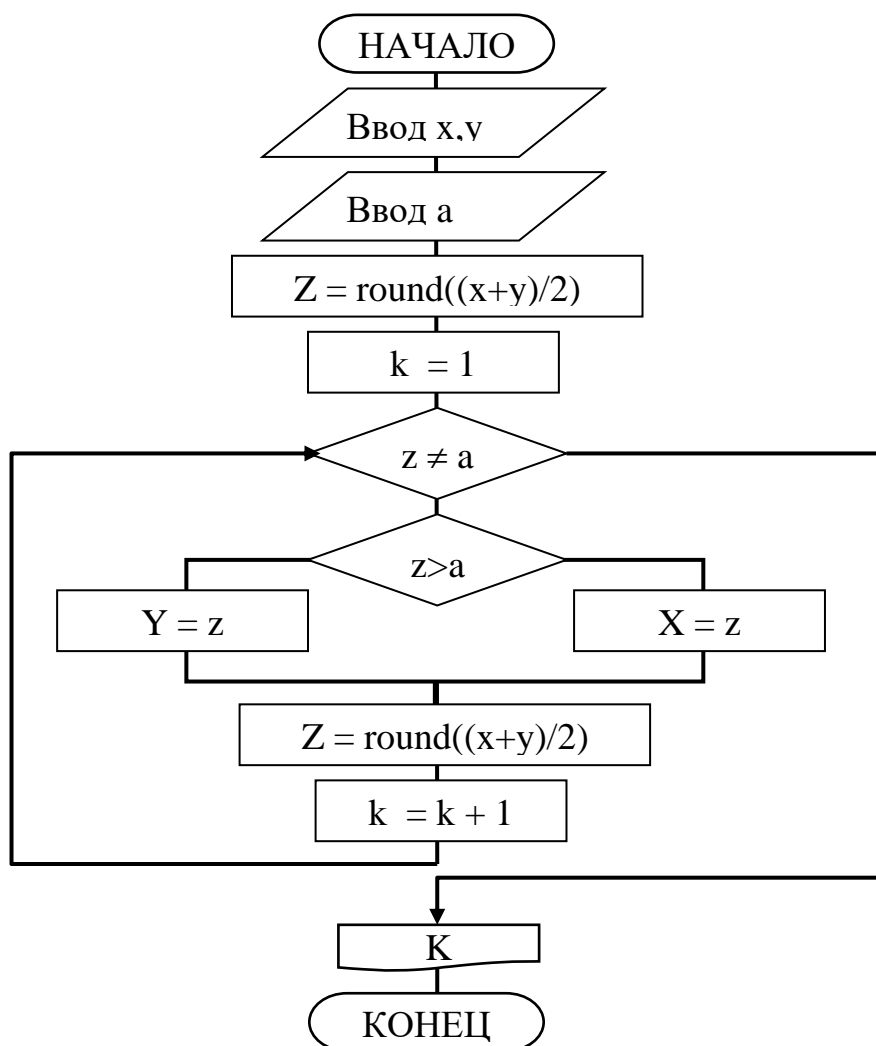
x, y – границы диапазона, целочисленные переменные;

a – задуманное число, целочисленная переменная;

z – вариант числа, предложенный Петей, целочисленная переменная;

k – количество попыток Пети, целочисленная переменная

Решение в виде блок-схемы (допускаются и другие блок-схемы алгоритмов, обеспечивающие вывод правильного ответа):



Решение на языке Turbo Pascal 7.0:

Program IGRA;

Var x,y,z,a,k: integer;

Begin

 Readln(x,y); //границы диапазона

 Readln(a); //задуманное число

 Z:=round((x+y)/2); //1-ый вариант числа, предложенный Петей

 k:=1; // количество попыток Пети

While z<>a **do**

begin

if z>a **then** y:=z **else** x:=z;

 Z:=round((x+y)/2); //очередной вариант числа Пети

 k:=k+1; // количество попыток Пети

End;

Writeln(k);

End.

Задание 4. Заготовка дров (100 баллов)

Тема: программирование.

Идея решения.

Решение данной задачи рассмотрим на примере теста из условия, когда распилов $X=18$, а получилось чурбачков $N=26$.

Пусть было A брёвен. С каждым распилом число чурбачков увеличивается на 1. Если сделать 1 распил, то число чурбачков $A+1$, если 2 – $A+2$, если 3 – $A+3$, ... 18 распилов – $A+18$ чурбачков.

Согласно тесту, получилось 26 чурбачков. Поэтому можно записать, что $A+18=26$, откуда $A=26-18=8$ (распилили 8 брёвен). Итоговая формула, по которой вычисляем купленное количество бревен: $A = N - X$

Количество бревен на каждой повозке в примере одинаково и равно 4. Если количество бревен нечетное, то первым выводим частное от целочисленного деления A на 2, а затем – тоже частное +1.

Решение на языке PascalABC.NET :

```
Var N1, N, X: integer;
```

```
begin
```

```
  assign(input, 'timber.in');
```

```
  reset(input);
```

```
  readln(x, n);
```

```
  close(input);
```

```
  assign(output, 'timber.out');
```

```
  rewrite(output);
```

```
  N1:=n - x;
```

```
  writeln(N1);
```

```
  if N1 mod 2 = 0 then write(N1 div 2, ' ', N1 div 2)
```

```
  else write(N1 div 2, ' ', N1 div 2+1);
```



```
close(output);
```

```
end.
```

Указания к проверке.

Напоминаем, что способ проверки задания 4 – это определение того, как предложенная участником программа соответствует требованиям условия задачи. Для проверки решения задания 4 Региональная предметно-методическая комиссия по информатике подготовила для муниципального жюри комплект тестов в электронном виде, представляющие собой текстовые файлы входных данных и соответствующие им файлы выходных данных, один из вариантов решения, а также проверяющая система (тестер), позволяющая для каждой задачи определять правильность полученного решения в автоматическом режиме. Все перечисленные выше материалы представлены на компакт-диске, который поступает в распоряжение организаторов муниципального этапа олимпиады до начала муниципального этапа.

Текст задания 4, предполагающего написание программы, содержит описание задачи, максимальное время работы программы на отдельном тесте, размер доступной программе памяти в процессе ее исполнения, форматы входных и выходных данных и примеры входных и выходных данных, варианты оценивания частичных решений.

Результатом решения задачи 4 является исходный текст программы на одном из разрешенных языков программирования. Допустимые языки и среды программирования указаны в файле «Материально-техническое обеспечение муниципального этапа».

С учетом сказанного, проверка решений участников осуществляется путем исполнения программы решения задания 4 с входными данными, соответствующими каждому тесту из предложенного Региональной предметно-методической комиссией по информатике комплекта тестов с

последующим анализом выходных файлов, получаемых в результате работы программы.

Поскольку участники олимпиады должны сдавать на проверку решения в виде исходного текста программы на одном из разрешенных языков программирования, то проверка решений каждого участника должна осуществляться в следующей последовательности:

- компиляция исходного текста программы;
- последовательное исполнение полученного exe-файла для файлов с входными данными, соответствующих тестам из набора тестов для данной задачи,
- сравнение результатов исполнения программы на каждом тесте с правильным ответом.

Перед компиляцией исходного текста программы жюри должно, просмотрев исходный код, убедиться, что сданное на проверку решение не работает ни с какими другими файлами, кроме входного (в режиме чтения) и выходного (в режиме записи). В случае нарушения этого требования программа участника не принимается к проверке и за неё начисляется 0 баллов.

При компиляции исходного текста программы, которую участник сдал на проверку, необходимо учитывать следующее.

1) Жюри должно использовать вполне определенные командные строки для компиляции решений, о чем участников следует известить до начала тура. Эту информацию необходимо разместить в Памятке участнику, например, в следующем виде:

Компилятор	Командная строка
Borland Delphi 7.0	dcc32 -cc <исходный файл>
Free Pascal 2.4.0	fpcc <исходный файл>
Visual C 2005	cl /O2 /TC <исходный файл>

GNU C 4.4.0 (MinGW)	gcc -O2 -x c -W1, -- stack=67108864 <исходный файл>
Visual C++ 2005	cl /O2 /EHs /TP <исходный файл>
GNU C++ 4.4.0 (MinGW)	g++ -O2 -x c++ -W1, -- stack=67108864 <исходный файл>
Visual Basic 2005	vbc <исходный файл>
Borland Pascal 7.0	bpc <исходный файл>
Borland C 3.1	bcc -ml <исходный файл>
Borland C++ 3.1	bcc -ml <исходный файл>

Жюри имеет право изменять команды компиляции решений в процессе проведения соревнований, но участники олимпиады должны быть обязательно проинформированы об этом перед началом тура.

2) Размер файла с исходным текстом программы не должен превышать 64 килобайта. Время компиляции программы не должно превышать 1 минуты. В случае нарушения этих ограничений решение участника считается неправильным и никакие баллы за эту задачу участнику не начисляются. Информация об этих ограничениях также должна быть размещена в Памятке участнику.

При исполнении программы на каждом тесте, в первую очередь, жюри должно определить, нарушаются ли заданные в условии этой задачи ограничения на время работы программы на отдельном тесте и размер доступной программе памяти в процессе ее исполнения. В случае нарушения названных ограничений баллы за этот тест участнику не начисляются. В случае применения тестовой системы контроль за ограничениями осуществляет данная система, которая выдает сообщения о нарушении временного ограничения, объема памяти и т.д. Только сообщение «ОК»

является сигналом о том, что тест пройден. Любое другое сообщение говорит о нарушении программой участника ограничений.

Если указанные выше ограничения в процессе исполнения программы с входными данными, соответствующими конкретному тесту, не нарушаются, то после завершения исполнения программы осуществляется проверка правильности полученного ответа. При проверке «вручную» осуществляется сравнение полученного выходного файла с файлом выходных данных из тестового набора данных.

В случае автоматической проверки с помощью тестовой системы, предоставленной региональной предметно-методической комиссией по информатике осуществляется подсчет количество тестов, получивших сообщение «ОК». За них начисляется количество баллов, указанное ниже для задачи.

Все представленные на проверку программы участников сначала должны проходить предварительное тестирование на тестах из условия задачи. Если на этих тестах решение участника выдает правильный ответ, то тогда это решение принимается жюри на окончательную проверку. Она осуществляется после завершения соответствующего тура на всех тестах из заданного набора тестов для этой задачи. В противном случае, решение участника считается неверным, и за него участнику не начисляются какие-либо баллы.

Самый простой, но самый трудоёмкий способ, - это вручную последовательно запускать проверяемую программу на каждом тесте из заданного комплекта и сравнивать результат программы участника с эталонным результатом. Для каждой задачи эти результаты имеются в файлах с расширением .A в комплектах тестов, предоставленных региональной предметно-методической комиссией по информатике на диске вместе с данным описанием.

Для автоматизации процесса проверки решений участников можно использовать либо разработанные своими силами командные файлы или простейшую программную систему автоматической проверки, либо использовать коммерческую программную систему для проведения олимпиад по информатике, либо одну из свободно распространяемых программных систем проведения олимпиад по информатике, информацию о которых можно найти в Интернете. Региональная предметно-методическая комиссия по информатике в каталоге tester080311 прилагает одну из таких свободно распространяемых программных систем проведения олимпиад TESTER build 080311 (файл !test.exe, автор Владимир Яковлев, 2003-2008, <http://acm.timus.ru/tester>). Эта программа является достаточно надежным и удобным средством для автоматической проверки решений на олимпиадах по программированию. Она используется для проверки решений участников Курского регионального этапа Всероссийской олимпиады школьников по информатике уже несколько лет и неплохо себя зарекомендовала. Согласно лицензионному соглашению, все авторские права на данную программу принадлежат только автору - Владимиру Яковлеву, но программу можно использовать бесплатно в некоммерческих целях, а также свободно распространять программу только вместе с лицензией, находящейся в отдельном файле readme.txt.

Материалы для непосредственной проверки решений задания 4 размещены на прилагаемом компакт-диске в отдельной папке с именем «timber», которая находится в папке «7-8 класс». Эта папка содержит:

- 1) Папку «preliminary», содержащую тесты из примеров в условии задачи и настроенную на данную задачу копию тестовой системы. Содержимое этой папки предназначено для предварительной проверки решений участника во время тура и после него. Данная папка должна быть сброшена на компьютеры участников до начала тура. Копируя в неё исполнимый файл своего решения и запуская !test.exe, участник осуществляет предварительную проверку своего решения во время тура.

Каждый тест из примера содержится в отдельном файле, входные файлы называются «timber01.in», «timber02.in». Файлы с правильными ответами называются «timber01.out», «timber02.out». Тесты пронумерованы в том же порядке, в котором они следуют в условии задачи.

2) Папку «tests», содержащую основные тесты для окончательной проверки и оценивания решений участников, правильные ответы. Каждый тест содержится в отдельном файле. Входные файлы называются «timber01.in», «timber02.in» и т.д. Файлы с правильными ответами называются «timber01.out», «timber02.out» и т.д.

3) Примеры правильного решения на языке Turbo Pascal 7.0 (файл sweets.pas), текст которого включен в ответы, содержащиеся в данном руководстве. Решения предоставляются только для ознакомления членов муниципального жюри с возможной реализацией правильного решения, а также для проверки работоспособности проверяющей системы. Их использование для генерации правильных ответов на тесты не требуется, так как соответствующие материалы уже содержат готовые правильные ответы на все тесты.

При проверке решений участников с использованием предоставленной тестирующей системы процесс предварительной проверки осуществляется в течение тура самими участниками. На компьютерах участников перед началом тура должна быть создана папка под именем «TIMBER». В неё должны быть сброшена папка «preliminary», соответствующая задаче. Участников нужно проинструктировать о следующем порядке работы с тестовой системой:

1) в папку «preliminary» скопировать исполнимый файл своего решения соответствующей задачи;

2) запустить на исполнение файл !test.exe. Если на тест выдается сообщение «ОК», то тест соответствует эталону, приведенному в условии.

Каждый тест из примера содержится в отдельном файле. Входные файлы называются «timber01.in», «timber02.in». Файлы с правильными ответами называются «timber01.out», «timber02.out». Тесты пронумерованы в том же порядке, в котором они следуют в условии задачи. Нужно также обратить особое внимание участников на то, что на тестирование должен копироваться файл с указанным в задаче именем (timber.exe). Эту информацию также следует разместить в Памятке участнику.

На усмотрение жюри остается вопрос о проверке программ участников, не использующих файловый ввод-вывод. При этом остается только «ручная» проверка по тому же комплекту тестов. Для этого варианта проверки в указаниях к проверке каждой задачи указаны номера тестов, которые реально проверить вручную. При этом за них начисляются те же баллы, что и при автоматической проверке. Дополнительно снимать баллы за отсутствие работы с файлами не нужно, т.к. все оставшиеся тесты будут считаться непройденными и участник даже за верное решение не сможет получить максимум баллов.

Описание тестовой системы

В случае, если имена входного-выходного файлов в решении участника не совпадают с заданными в условии, жюри может изменить файл конфигурации тестовой системы. Нужно открыть в блокноте этот файл !test.ini и настроить его на работу с проверяемой программой. Имена входного-выходного файлов программы задаются в двух последних строках – это параметры InputFile и OutputFile. Также возможно изменить имя проверяемой программы – параметр RunProgram. Пример файла конфигурации задания 1 для 9-11 классов (жирным выделены параметры, которые уже настроены под проверяемую задачу):

RunProgram	= election.exe
TestInput	= 00.in
TestAnswer	= 00.out

```

TestNumber      = Auto
TimeLimit      = 1.0
MemoryLimit    = 64.0
Checker        = #token
HideCheckerOutput = No
HideProgramOutput = Yes
GenAnswers     = Auto
Interrupt      = No
Pause         = Yes
InputFile     = election.in
OutputFile    = election.out

```

```

; === Краткая помощь (более детально читайте файл readme.txt) ===
;RunProgram      Исполнимый файл участника для тестирования
;TestInput       Маска имени входного файла. Последние '00' или
;                '000' будут заменяться на текущий номер теста
;TestAnswer      Маска имени выходного файла. Mask of test answer file.
;                '000' будут заменяться на текущий номер теста
;TestNumber      Количество тестов или AUTO для автоматического
;                определения этого количества
;TimeLimit       Максимальное время в сек. для работы программы
;MemoryLimit     Максимальный размер памяти, разрешенный
программе
;                или NO, чтобы убрать ограничение памяти
;Checker         Программа проверки, используйте #TOKEN
;HideCheckerOutput Если значение этого параметра равно YES или 1,
;                то вывод проверяющей программы подавляется, и на
;                экран выводится только либо "ok" либо "wrong
;                answer"
;HideProgramOutput Если значение этого параметра равно YES или 1,
;                то консольный вывод проверяемой программы подавляется.
;GenAnswers      Если значение этого параметра равно YES или 1, то
;                содержимое текущих файлов с выходными данными будет
;                заменено результатами работы программы на каждом тесте.
;Interrupt       Если значение этого параметра равно YES или 1, то работа
;                Тестера завершается после первого неудачного запуска.
;                NO или 0 - продолжение тестирования в любом случае
;                (рекомендуемое)
;Pause          Если значение YES или 1, то по окончании работы Тестера

```



```

;   происходит задержка экрана. NO или 0 - работа завершается
;   немедленно
;InputFile  Файл, из которого тестируемая программа будет читать
;   входные данные (имя входного файла).
;OutputFile Файл, в который тестируемая программа будет выводить
;   результаты работы (выходной файл).
;ScorePerTest  Баллы, начисляемые за каждый пройденный тест.
;   Необязательный параметр. Значение по умолчанию - 0.
;ScoreAccepted Баллы, начисляемые за прохождение всех тестов.
;   Необязательный параметр. Значение по умолчанию –
;   ScorePerTest, умноженное на количество тестов. В случае
;   отсутствия обоих параметров, баллы не подсчитываются.

```

Характеристика тестов для задания 4

Комплект тестов разрабатывался таким образом, чтобы имелась возможность в максимальной степени оценить решения участников. Количество тестов – 20. Файлы с тестами находятся в каталоге tests. В общем случае в комплекте тестов выделяются следующие группы тестов:

- 1) простые тесты;
- 2) тесты на частные случаи, позволяющие выявить особенности используемых алгоритмов (например, используемые типы данных);
- 3) общие тесты (достаточно случайные тесты, разные по размеру: от простых тестов до сложных);
- 4) тесты максимальной размерности (тесты с использованием максимальных значений входных переменных, позволяющие оценить эффективность предложенных алгоритмов или их работоспособность при максимальной размерности задачи).

Система оценивания решений участников

Система оценивания решений заданий для 7-8 классов основана на следующих положениях:

1. Решение задания 1 оценивается из 40 баллов – это максимальное количество баллов, которое участник может получить за полное решение задания 1.

2. Решение задания 2 оценивается из 60 баллов – это максимальное количество баллов, которое участник может получить за полное решение задания 2.

3. Решение задания 3 оценивается из 100 баллов – это максимальное количество баллов, которое участник может получить за полное решение задания 3.

4. Решение задания 4 оценивается из 100 баллов – это максимальное количество баллов, которое участник может получить за полное решение задания 4.

5. Общая оценка за решение задания 4 конкретным участником складывается из суммы баллов, начисленных ему по результатам исполнения всех тестов из предложенного для этой задачи набора. Каждый из предлагаемых 20 тестов оценивается одинаково – по 5 баллов.

6. Итоговый результат каждого участника подсчитывается как сумма полученных этим участником баллов за решение каждой задачи. С учетом того факта, что предлагается четыре задачи по 40, 60, 100 и 100 баллов, то максимально возможное количество баллов, которое может набрать участник по итогам муниципального этапа, составляет 300 баллов.