

**Ответы и краткие методические рекомендации по
проверке заданий муниципального этапа Всероссийской
олимпиады школьников по информатике и ИКТ
9-11 классы**

**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.

2. Задания для 9 – 11 классов предполагают решения в виде компьютерной программы, представленной на проверку в виде файла с исходным кодом на одном из разрешенных языков программирования.

Задание 1. Игра в числа (100 баллов)

Тема: программирование разветвляющихся алгоритмов

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.

Идея решения данной задачи. Представим искомое число как ABCDEF. Поскольку каждая следующая цифра на 1 меньше предыдущей, то имеем арифметическую прогрессию, разностью которой равна 1.

Тогда цифры искомого числа обладают следующим свойством $A+F=B+E=C+D$. Из этого следует, что N должно быть кратно 3 и N ограничено числами $(9+8+7+6+5+4) \leq N \leq (5+4+3+2+1+0)$.

Учитывая, что C и D это идущие подряд натуральные числа их сумма всегда нечетна. Поэтому частное от деления N на 3 должно быть нечетным.

Решение на языке PascalABC.NET :

```

var
  N, X, x1: integer;
begin
  assign(input, 'numeric.in');
  reset(input);
  readln(n);
  assign(output, 'numeric.out');
  rewrite(output);
  if (n mod 3 <> 0) or (n > 39) or (n < 15)
    then write('impossible')
  else
    begin
      x := n div 3;
      if x mod 2 = 1 then
        begin
          x1 := x div 2;
          write(x1+3, x1+2, x1+1, x1, x1-1, x1-2);
        end
      else write('impossible');
    end;
  close(output);
  close(input);
end.

```

Задание 2. Системы счисления (100 баллов)

Тема: Программирование циклических алгоритмов.

Решение данной задачи основано на применении стандартного алгоритма перевода чисел из десятичной системы счисления в систему с основанием Q .

Решение на языке PascalABC.NET :

```

program notation;
var q, x: integer;
    res, c: string;

```

```

begin
  assign(input, 'notation.in');
  reset(input);
  res:='';
  Readln(q, x);
  while x>0 do
    begin
      str(x mod q, c);
      res:=c+res;
      x:=x div q;
    end;
  close(input);
  assign(output, 'notation.out');
  rewrite(output);
  writeln(res);
  close(output);
end.

```

Задание 3. Рисунки на полях (100 баллов)

Тема: обработка линейного потока данных.

Идея решения.

Нам нужно определить площадь треугольника, одно из оснований которого лежит на оси ОХ. Пусть длина этого основания равна В, а высота треугольника равна Н. Тогда площадь треугольника равна $S = B \cdot H / 2$.

Из этих рассуждений следует, что треугольник с максимальной площадью образован двумя точками на оси ОХ, которые дальше всего отстоят друг от друга, и точкой, которая наиболее удалена от оси ОХ, то есть имеет максимальную координату у (по модулю).

Таким образом, нужно найти:

1. xMin – минимальное значение х-координаты среди всех точек, для которых у-координата равна нулю;
2. xMax – максимальное значение х-координаты среди всех точек, для которых у-координата равна нулю;
3. yMax – максимальный модуль у-координаты среди всех точек.

Тогда $S = (xMax - xMin) \cdot yMax / 2$.

За начальные значения x_{Max} и x_{Min} принимаем значения $-\text{MaxInt}$ и MaxInt соответственно. MaxInt – это максимально возможное значение для типа `integer`, который использован при решении.

Решение на языке PascalABC.NET :

```

program triangles;
var i, N: integer;
    minX, maxX, maxY, x, y: real;
    f: boolean;
begin
    assign(input, 'triangles.in');
    reset(input);
    minX := maxint; maxX := -maxint; maxY := 0;
    Readln(n);
    f:=false;
    for i:=1 to N do
    begin
        Readln(x, y);
        if y = 0 then
        begin
            f:=true;
            if (x < minX) then minX:= x;
            if (x > maxX) then maxX:= x;
        end
        else if abs(y) > abs(maxY) then maxY:=y;
    end;
    close(input);
    assign (output, 'triangles.out');
    rewrite(output);
    if f then writeln(((maxX - minX)*abs(maxY)/2):0:0)
    else writeln(0);;
    close (output);
end.

```

Задание 4. Связь (100 баллов)

Тема: Перебор, геометрические алгоритмы.

Идея решения.

Формализуем предметную область. Так как в условии задачи требуется найти координаты расположения штаба с учетом того, чтобы было минимальным время, через которое все подразделения получают круговой сигнал, отправленный из штаба, то круговой сигнал – это окружность, а мин

время доставки сигнала (учитывая что скорость распространения одинаковая и равна 1 км/с) - это минимальный радиус окружности, включающей в себя все точки, при условии, что штаб находится в центре данной окружности.

Как известно любые три точки определяют окружность. Минимальной будет окружность, включающая в себя все точки плоскости, которая построена по таким трем точкам, что сумма расстояний между ними максимальна. Необходимо применить типовой алгоритм формирования групп сочетания без повторов и вычисление радиуса описанной окружности. Отдельно необходимо рассмотреть случай, когда все входные точки лежат на 1 прямой.

Решение на языке PascalABC.NET :

```

var x, y: array [1..1000] of integer;
    n, i, j, r, num1, num2, num3: integer;
    ras1, ras2, ras3, max: int64;
    ma, mb, x0, y0, rad: real;
function max3(a, b, c: integer): integer;
var max: integer;
begin
    max:=a;
    if max<b then max:=b;
    if max<c then max:=c;
    max3:=max;
end;
function min3(a, b, c: integer): integer;
var min: integer;
begin
    min:=a;
    if min>b then min:=b;
    if min>c then min:=c;
    min3:=min;
end;

begin
    assign(input, 'comm.in');
    reset(input);
    readln (n);
    for i:=1 to n do readln (x[i], y[i]);
    max:=0;
    for i:= 1 to (n - 2) do
        for j:= i + 1 to (n - 1) do

```

```

for r:= j + 1 to n do
begin
  ras1:=sqr(x[r]-x[j])+sqr(y[r]-y[j]);
  ras2:=sqr(x[j]-x[i])+sqr(y[j]-y[i]);
  ras3:=sqr(x[i]-x[r])+sqr(y[i]-y[r]);
  if (ras1+ras2+ras3)>max then
    begin
      max:=(ras1+ras2+ras3);    num1:=i; num2:=j;
num3:=r;
    end;
  end;
  assign(output, 'comm.out');
  rewrite(output);
  if abs((x[num3]-x[num1])*(y[num2]-y[num1])-(x[num2]-
x[num1])*(y[num3]-y[num1]))<0.000000001
  then
    begin //точки на прямой
      if (x[num1]=x[num2]) and (x[num2]=x[num3]) then
//координаты X всех точек равны
      begin
        x0:=x[num1];
        y0:=(max3(y[num1], y[num2], y[num3]) -
min3(y[num1], y[num2], y[num3])) div 2;
      end
      ELSE
      begin
        x0:=(max3(x[num1],x[num2],x[num3]) -
min3(x[num1],x[num2],x[num3])) div 2 +
min3(x[num1],x[num2],x[num3]);
        y0:=(max3(y[num1], y[num2], y[num3]) -
min3(y[num1], y[num2], y[num3])) div 2 + min3(y[num1],
y[num2], y[num3]);
      end;
      rad:=sqrt(sqr(x0-x[num1])+sqr(y0-y[num1]));
      writeln(output,round(rad));
      writeln(output,round(x0), ' ', round(y0));
    end
  else
  begin
    //центр
    ma:=(y[num2]-y[num1])/(x[num2]-x[num1]);
    mb:=(y[num3]-y[num2])/(x[num3]-x[num2]);
    x0:=(ma*mb*(y[num1]-y[num3])+mb*(x[num1]+x[num2]) -
ma*(x[num2]+x[num3]))/(2*(mb-ma));

```

```

y0:=-1/ma*(x0-
(x[num1]+x[num2])/2)+(y[num1]+y[num2])/2;
//радиус
rad:=sqrt(sqr(x0-x[num1])+sqr(y0-y[num1]));
writeln(output,round(rad));
writeln(output,round(x0),' ',round(y0));
end;
close(output);
end.

```

Характеристика тестов задач

Комплект тестов разрабатывался таким образом, чтобы имелась возможность в максимальной степени оценить решения участников. Количество тестов – 20. Файлы с тестами находится 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

Решение всех заданий оценивается из 100 баллов – это максимальное количество баллов, которое участник может получить за полное решение. Общая оценка за решение каждого задания конкретным участником складывается из суммы баллов, начисленных ему по результатам исполнения всех тестов из предложенного для этой задачи набора.

№ задачи	Название	Количество тестов	Балл за один тест
1	Игра в числа	25	4

2	Системы счисления	50	2
3	Рисунки на полях	20	5
4	Связь	20	5

Итоговый результат каждого участника подсчитывается как сумма полученных этим участником баллов за решение каждой задачи. С учетом того факта, что предлагается четыре задачи по 100 баллов, то максимально возможное количество баллов, которое может набрать участник по итогам муниципального этапа, составляет 400 баллов.

Для проверки решений участников, а также для проведения олимпиады Региональная предметно-методическая комиссия по информатике предлагает использовать сервис Яндекс.Контест. Это сервис для онлайн-проверки заданий по математике и программированию. Яндекс.Контест поддерживает более двадцати языков программирования. Региональная предметно-методическая комиссия по информатике подготовила и разместила с помощью данного сервиса все необходимые материалы для проведения олимпиады. Важнейшее условие для проведения олимпиады – на компьютерах участников должен быть заблокирован выход в Интернет, кроме выхода на сайт проверяющей системы - <https://contest.yandex.ru/>.

Во время проведения тура свои решения участники проверяют автоматически средствами настроенного соревнования в сервисе Яндекс.Контест с помощью набора тестов, составленных Региональной предметно-методической комиссией по информатике. Участники отправляют свои решения и через некоторое время могут видеть результаты проверки. Сервис способен одновременно обрабатывать терабайты данных, поэтому выдерживает нагрузку в более чем тысячу участников.

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

При компиляции исходного текста программы проверяющая система использует 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>

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.

Во время тура все представленные на проверку программы участников сначала проходят предварительное тестирование на тестах из условия задачи. В случае правильного ответа решение принимается 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.

В случае невозможности использования сервиса Яндекс.Контест Для проверки решений можно использовать программу тестер, размещенную на прилагаемом компакт-диске в папке с именами «timber», которая находится в папке «7-8 класс». Тестер настроен на проверку 4 задачи для 7-8 классов, поэтому необходимо изменить файл инициализации согласно прилагаемой инструкции.

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.