

## Второй (заключительный) этап академического соревнования

### Олимпиады школьников «Шаг в будущее»

#### по общеобразовательному предмету «Информатика»

10 класс, февраль, 2016 г.

#### Вариант № 1.

#### Задание 1 (12 баллов)

Определить минимальное основание системы счисления, для которого существует решение уравнения  $4_y * x_y + 2_y * x_y = 20_y$ .  $X$  - целое число.

Ответ: 6

Решение. Перепишем уравнение в виде  $6x=2y$ . Очевидно, что  $3x=y$ , следовательно, основание системы счисления кратно 3. 3 нам не подходит, так как в троичной системе счисления нет цифры 4. Следовательно, ответ 6.

Критерии оценки.

На полный балл оценивается решение, в котором в явном виде приведено уравнение или цепочка рассуждений, ведущая к обоснованию правильного ответа.

Решение, которое содержит арифметические ошибки, не ведущие к нарушению цепочки рассуждений, оценивается в 8 баллов.

Решение, содержащее только правильный ответ, оценивается в 3 балла.

#### Задание 2 (12 баллов)

На одном званом вечере среди гостей оказалось пять офицеров: пехотинец, артиллерист, лётчик, связист и сапёр. Один из них - капитан, трое - майоры, один - полковник. Дамы окружили офицеров таким вниманием, что все остальные гости оказались просто забытыми. Из разговора удалось выяснить следующее:

- у Петра такое же звание, как и у его друга сапёра;
- офицер-связист и Николай - большие друзья;
- офицер-лётчик вместе с Владимиром и Александром недавно были в гостях у Николая;
- незадолго до званого вечера у артиллериста и сапёра почти одновременно вышли из строя радиоприёмники. Оба обратились к Александру с просьбой зайти к ним и помочь связисту устранить неисправность. С тех пор приёмники у обоих работают отлично.
- Николай чуть было не стал лётчиком, но потом по совету своего друга сапёра избрал иной род войск
- Пётр по званию старше Александра, Владимир по званию старше Николая. (Звания по старшинству от младшего к старшему: капитан, майор, полковник).

Определите звание каждого офицера, и род войск в котором он служит.

Критерии:

- -5 баллов за каждое несоответствие

Решение:

1. Так как Александр не артиллерист, не сапер, не связист, не летчик (это следует из условий 3,4), значит от пехотинец.
2. Т.к. Николай не летчик, не сапер, не связист (из п. 2,5 ), значит от артиллерист.
3. Из условия 1 следует, что Петр может быть только майором. Т.к. Петр (майор) по званию старше Александра (по условию 6), значит, Александр - капитан
4. Т.к. Владимир по званию старше Николая, по условию 6, а Николай не капитан значит Владимир полковник, а Николай майор.
5. Остается что Андрей майор.
6. Так как сапер майор, но не Петр (см п.1) и не Николай и не Александр и не Владимир, значит сапер Андрей.
7. Из п.3 следует, что Владимир не летчик, значит от связист, а Петр летчик

	Пехотинец	Летчик	Артиллерист	Связист	Сапер	Капитан	Майор	Полковник
Петр	-	+	-	-	--	-	+	-
Николай	-	-	+	-	-	-	+	-
Владимир	-	-	-	+	-	-	-	+
Александр	+	-	-	-	-	+	-	-
Андрей	-	-	-	-	+	-	+	-

### Задание 3 (12 баллов)

В 11 классе учится 30 человек. Известно, что ЕГЭ по химии будут сдавать 15 человек. ЕГЭ по информатике будет сдавать 13 человек. Химию и информатику, но не физику, сдают 8 человек. Физику и информатику, но не химию, сдают 5 человек. Химию и физику, но не информатику, сдают 6 человек. Сколько человек сдаёт физику, если известно, что каждый ученик сдаёт хотя бы один предмет из перечисленных?

Ответ: 21

Решение.

Очевидно, что если 8 человек сдают и химию, и информатику, но не физику, а 5 человек сдают и физику, и информатику, но не химию, но при этом всего 13 человек сдают информатику, то и

физику, и химию, и информатику сдает 0 человек. Следовательно, пересечение трех множеств пустое.

Значит, если обозначить количество сдающих физику за  $X$ , получаем:

$$X+15+13-5-6-8=30, X+9=30, X=21.$$

Критерии. На полный балл (12 баллов) оценивается решение, в котором в явном виде приведены круги Эйлера или цепочка рассуждений, ведущая к обоснованию правильного ответа.

Решение, которое содержит арифметические ошибки, не ведущие к нарушению цепочки рассуждений, оценивается в 8 баллов.

Решение, содержащее только правильный ответ, оценивается в 5 баллов во всех классах.

#### Задание 4 (12 баллов)

На вход автомату подаётся четырёхзначное число в шестнадцатеричной системе счисления.

Автомат выполняет следующие действия:

- переводит число в двоичную систему счисления;
- инвертирует разряды числа;
- переводит число в шестнадцатеричную систему счисления.

Если на вход подать не четырёхзначное число - автомат работать не будет.

Какое максимальное число можно получить в результате работы автомата.

Критерии (макс. 15 баллов)

- -8 если ответ не в той системе счисления
- -10 если ответ EEEE
- -12 если ответ 9999
- -5 нет решения

Решение:

1. Если мы подадим FFFF, то получим 0000, т.е. 0. следовательно в старей тетраде в изначальном числе должен быть один 0. и максимальное число будет EFFF

#### Задание 5 (12 баллов)

Помещик нанял двух крестьян и обещал по окончании работы дать каждому по 5 мер овса. Когда работа была окончена, помещик велел отдать в распоряжение работавших крестьян 3 мешка: один мешок с 10 мерами овса, а два других, вместимостью 7 мер и 3 меры, пустые. Других мешков или других ёмкостей у крестьян не было, однако они разделили овёс так, что каждый унёс домой по 5 мер овса. Как крестьяне произвели этот делёж?

Решение

Операция	Мешок		
	10 мер	7 мер	3 меры
Первоначально	10	0	0

1-е пересыпание	$10-3=7$	0	$0+3=3$
2-е пересыпание	7	$0+3=3$	$3-3=0$
3-е пересыпание	$7-3=4$	3	$0+3=3$
4-е пересыпание	4	$3+3=6$	$3-3=0$
5-е пересыпание	$4-3=1$	6	$0+3=3$
6-е пересыпание	1	$6+1=7$	$3-1=2$
7-е пересыпание	$1+7=8$	$7-7=0$	2
8-е пересыпание	8	$0+2=2$	$2-2=0$
9-е пересыпание	$8-3=5$	2	$0+3=3$
10-е пересыпание	5	$2+3=3$	$3-3=0$

Критерии

Оценивается только решение, приведшее к правильному результату.

Если количество операций превышает количество операций эталонного решения, то вычитается 3 балла за каждую лишнюю операцию.

### Задание 6 (15 баллов)

Вася собирается на выходных поиграть в “Лигу Легенд” вместе со своими одноклассниками. Поскольку Вася хочет играть с хорошей командой, он спросил одноклассников в скайп-конференции про их успехи за прошедший месяц. Для игры ему необходимо ещё четыре члена команды. Требуется найти игроков, с которыми захочет играть Вася.

Формат входных данных:

На первой строке задаётся число  $N$  - количество ответов, которые Вася получил в скайп-конференции.  $N$  больше или равно 4.

В следующих  $N$  строках в квадратных скобках записано имя ответившего одноклассника, затем стоит двоеточие и пробел, после чего идёт ответ. Ответ гарантированно содержит ровно одно целое неотрицательное число, ограниченное пробелом или концом строки. Это число является количеством побед одноклассника за прошедший месяц. В ответе не встречается квадратных скобок.

Вам необходимо определить имена четырёх одноклассников Васи, набравших больше всего побед, и вывести их на экран в произвольном порядке.

**Пример:**

*Входные данные:*

5

[Вася]: У меня 5 побед.

[Коля]: Я выиграл 10 игр.

[Маша]: Взяла 7 игр.

[Олег]: 2

[Петя]: Очень плохо. 15 побед только.

*Выходные данные:*

Вася

Коля

Маша

Петя

Критерии оценки. В 15 баллов оценивается верно работающее решение задачи, удовлетворяющее следующим критериям:

- время работы программы пропорционально количеству входных данных;
- входные данные не хранятся в массиве целиком

Не оптимальное по расходу памяти решение (например, записать все в массив и найти 4 максимума) оценивается в 11 баллов.

Не оптимальное по времени работы решение (например, записать все в массив и отсортировать его) оценивается в 7 баллов.

Неверно работающая программа, из которой, тем не менее, понятно, что участник понимает решение задачи, оценивается в 3 балла.

{будем по очереди читать каждую строчку

поскольку точно известен формат входных данных, то

будем извлекать очередное число и вставлять его

в массив, где хранятся 4 лучших результата.

Массив будем держать отортированным, по сути выполняя вставку

}

```
program league;
```

```
const M=4; //нам надо найти 4 игроков
```

```
type tmas=array [1..M] of integer; //тип массива для побед
```

```
    tlines=array[1..M] of string; //тип массива для имен игроков
```

```
function cut(var s:string):string;
```

```
var res:string;
```

```
begin
```

```
    res:=copy(s,1,pos(':',s)-1); //копируем кусок строки внутри квадратных скобок
```

```
    delete(s,1,pos(':',s)+2); //удаляем имя и скобки из строки
```

```
    delete(res,1,1); //удаляем из имени открывающую скобку
```

```
    cut:=res;
```

```
end;
```

```
function win(s:string):integer;
```

```
var flag:boolean;
```

```

x,code:integer;
num:string;
begin
s:=s+' ';
while pos(' ',s)>0 do
delete(s,pos(' ',s),1); //чистим лишние пробелы
flag:=false; //flag отмечает, найдено ли число.
while not flag do //Оно гарантированно есть и проверять наличие слов в строке поэтому не
надо
begin
num:=copy(s,1,pos(' ',s)-1);//извлекаем слово
delete(s,1,pos(' ',s)); //удаляем слово
val(num,x,code); //пытаемся преобразовать слово в число
if code = 0 then //если преобразование удачно, то поднимаем флаг выхода
begin
flag:=true;
end;
end;
win:=x;
end;
procedure select(var lines:tlines;var mas:tmas; name:string; x:integer);
var i,n:integer;
begin
n:=4;
while (mas[n]>x) and (n>0) do //идем по массиву в поисках места для числа
n:=n-1;
if n>0 then //если число можно вставить в массив, то
begin
for i:=2 to n do
begin
mas[i-1]:=mas[i]; //сдвигаем массивы и побед, и имен
lines[i-1]:=lines[i];
end;
lines[n]:=name;
mas[n]:=x;
end;
end;
var n,i:integer;

```

```
mas:tmas;  
names:tlines;  
s,name:string;
```

```
begin
```

```
readln(n); //вводим количество записей
```

```
for i:=1 to M do
```

```
mas[i]:=-1; //инициализируем массив так, чтобы вставка была возможна с гарантией
```

```
for i:=1 to n do
```

```
begin
```

```
readln(S); //читаем очередную строчку
```

```
name:=cut(S); //извлекаем имя и оставляем в строке только текст сообщения
```

```
select(names,mas,name,win(s)); //проверяем, можно ли вставить в массив число побед
```

```
end;
```

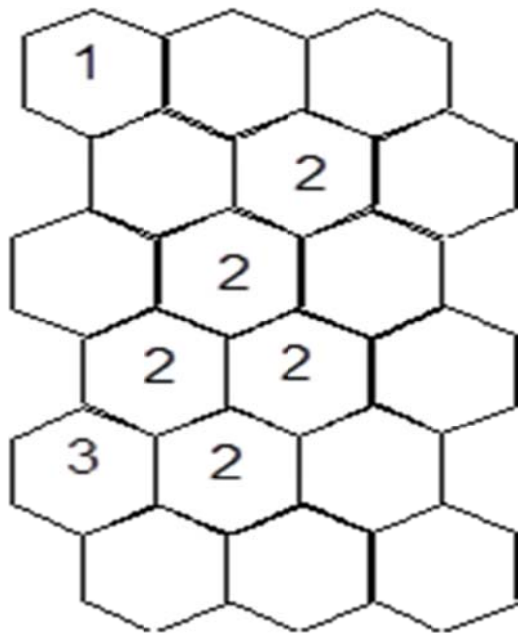
```
for i:=1 to M do //выводим результат. Так как игроков по условию задачи не меньше четырех,  
имена будут всегда и проверять ничего не надо
```

```
writeln(names[i]);
```

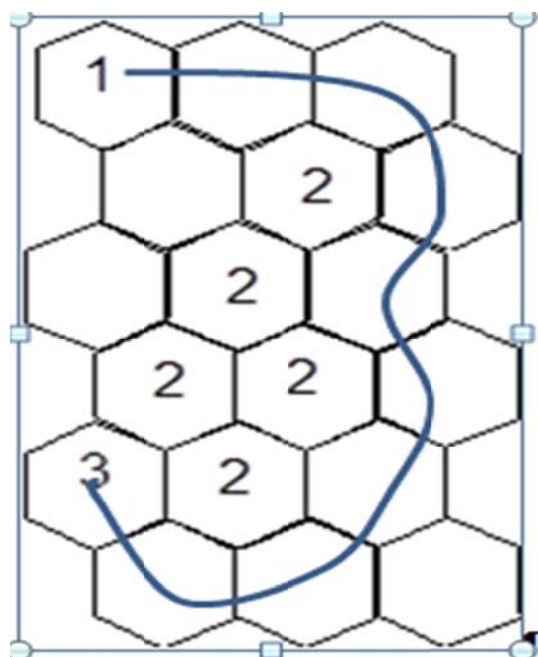
```
end.
```

### Задание 7 (25 баллов)

Пусть имеются пчелиные соты (см. рис.). Пчеле (на рис. обозначена цифрой **1**) необходимо попасть в ячейку обозначенную цифрой **3**. Цифрой **2** обозначается ячейка занятая другой пчелой.



Написать программу, которая найдёт кратчайший путь. Известно, что пчела может перемещаться только в соседнюю ячейку. Пчела не может заходить в ячейки, занятые другими пчёлами (количество рядов в сотах не более 20, количество ячеек в ряду не более 40).



В качестве ответа программа должна вывести на экран количество ячеек, которые вошли в этот кратчайший путь (начальная и конечная ячейка учитываются)

Исходные данные находятся в текстовом файле. В первой строке через пробел записаны два целых числа  $N$  – количество рядов и  $M$  – количество ячеек в ряду. Далее записано  $N$  строк. В каждой строке записано  $M$  цифр без пробелов. Цифры означают следующее:

0 – ячейка пустая;

1 – в этой ячейке находится пчела, которая должна

пройти искомый путь;

2 – в этой ячейке находится пчела;

3 – ячейка, в которую необходимо построить путь

Для вышеописанного примера файл выглядит следующим образом:

```
6 3
```

```
100
```

```
020
```

```
020
```

```
220
```

```
320
```

```
000
```

```
Ответ
```

```
10
```

Критерии

Правильно работающее решение задачи оценивается в 25 баллов.

Если программа работает, но находит не оптимальный, а некий другой путь, который не нарушает остальных правил, то такая программа оценивается из 20 баллов.

За ввод с клавиатуры (вместо файла) вычитается 2 балла. Синтаксические ошибки, которые не влияют на логику работы программы, не учитываются (например, `begun`, `than`, пропущенные точка с запятой и т.п.). Если программа работает в целом правильно, но есть ошибки типа `or` вместо `and`, `больше` вместо `меньше` и наоборот и т.п., то такие ошибки наказываются вычитанием по 2 балла за каждую, но не более 3-х ошибок такого рода. Если таких ошибок больше, то считается, что в целом такая работа работает неправильно.



Если программа в целом работает не правильно, то оцениваем отдельные части программы, а именно

1. Если программа считывает данные и создаёт адекватную структуру, например, массив, в котором фиксируются все необходимые данные (место нахождения пчёл, начальное и конечное положение пчелы), то данная часть программы оценивается в 5 баллов
2. Если программа правильно выполняет волну (или другой оптимальный поиск), то эта часть программы оценивается в 10 баллов
3. Если программа может строить правильный, но не оптимальный путь, то эта часть программы оценивается в 5 баллов.

{Идея решения

Считываем данные из файла в массив, отмечая специальными числами положение пчёл, начало и конец пути. Затем запускаем классическую волну, но учитываем, что каждая ячейка связана не с 4 ячейками (как в обычном двумерном массиве), а с шестью. Учитываем, что для чётных и нечётных рядов связи различны.}

**const**

row = 10;

col = 20;

**type**

mas=array[1..row, 1..col] of integer;

**procedure** init(var a: mas);

**var**

i, j: integer;

**begin**

**for** i := 1 **to** row **do**

**for** j := 1 **to** col **do**

a[i,j]:=0

**end**;

**procedure** inp(var f: text; var a: mas; n,m: integer);

**var**

i,j,k,code: integer;

s: string;

**begin**

**for** i := 1 **to** n **do** //перебираем все строчки

**begin**

readln(f, s); //считываем строку с ячейками

**for** j:=1 **to** m **do** //перебираем ячейки в строке

**begin**

val(copy(s,j,1),k,code);

```

case k of
0,1:a[i,j]:=k;
2:a[i,j]:=-1;//пчёл отметим числами -1
3:a[i,j]:=-100;//конечный пункт -100
end
end;
end;
end;
procedure make(var x:integer;hod:integer;var Ok:boolean);
begin
case x of
-100: begin //если конечный пункт, то
    Ok:=true; //добрались
    x:=hod
    end;
0: x:=hod; //если ячейка пустая, то делаем в неё ход
end
end;
procedure sosedi(var a:mas;i,j,n,m,hod:integer;var Ok:boolean);
begin
if j>1 then //если слева есть ячейка, то
    make(a[i,j-1],hod,Ok); //пытаемся сделать туда ход
if j<m then //если справа есть ячейка, то
    make(a[i,j+1],hod,Ok);//пытаемся сделать туда ход
if i mod 2 =0 then //если ряд чётный
begin
    make(a[i-1,j],hod,Ok);//пытаемся сделать ход в ячейку сверху слева
if j<m then //если справа сверху есть ячейка, то
    make(a[i-1,j+1],hod,Ok);//пытаемся сделать ход в ячейку сверху справа
if i<n then //если снизу есть ряд, то
begin
    make(a[i+1,j],hod,Ok);//пытаемся сделать ход в ячейку снизу слева
if j<m then //если справа снизу есть ячейка, то
    make(a[i+1,j+1],hod,Ok);//пытаемся сделать туда ход
end
end
end
else
begin

```

```

if i>1 then //если сверху есть ряд,то
begin
  make(a[i-1,j],hod,Ok);//пытаемся сделать ход в ячейку сверху справа
if j>1 then //если слева сверху есть ячейка, то
  make(a[i-1,j-1],hod,Ok);//пытаемся сделать ход в ячейку сверху слева
end;
if i<n then //если снизу есть ряд,то
begin
  make(a[i+1,j],hod,Ok);//пытаемся сделать ход в ячейку снизу справа
if j>1 then //если слева снизу есть ячейка, то
  make(a[i+1,j-1],hod,Ok);//пытаемся сделать туда ход
end
end
end;
procedure volna(a:mas; n, m:integer;var res:integer);// запускаем волну
var
  i,j,k:integer;
  OK:boolean;
begin
  res:=1;
  OK:=false;
repeat //повторяем
  for i:=1 to n do //перебираем все ячейки
  for j:=1 to m do
    begin
      if a[i,j]=res then //если в текущую ячейку попали на предыдущем ходе ,то
      sosed(a,i,j,n,m,res+1,Ok)// в соседние ячейки проставляем очередной номер хода
    end;
    inc(res);
  until OK; //пока не дойдем до конечной ячейки
end;
var
  n,m,res:integer;
  soty:mas;
  f:text;
begin
  Assign(f, 'vvod.txt');

```