

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 отсортированным, 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;
```

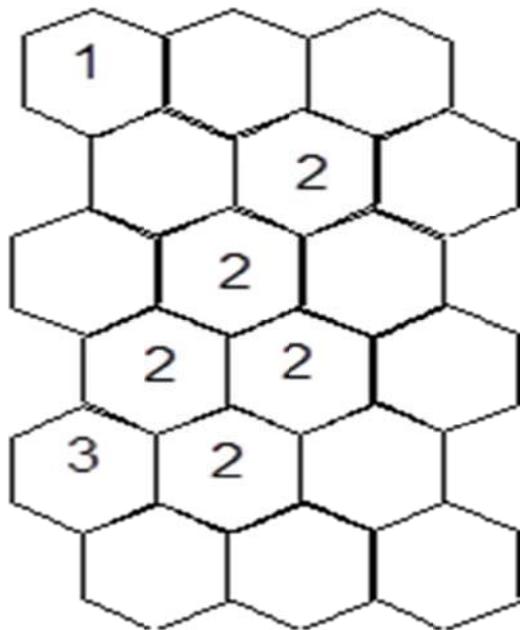
```
for i:=1 to M do //выводим результат. Так как игроков по условию задачи не меньше четырех,  
имена будут всегда и проверять ничего не надо
```

```
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).

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');

```

Второй (заключительный) этап академического соревнования

Олимпиады школьников «Шаг в будущее»

по общеобразовательному предмету «Информатика»

10 класс, февраль, 2016 г.

Вариант № 2.

Задание 1 (12 баллов)

Определить основание системы счисления, в которой записано выражение:

$$\begin{array}{r} aba_y \\ + b4_y \\ \hline b00_y \end{array}$$

где a и b - цифры числа.

Ответ: 5

Решение. Исходя из формулы, запишем систему уравнений.

$$a+4=y$$

$$b+b+1=y$$

$$a+1=b$$

Очевидно, что $a+4=2a+3$, следовательно, $a=1$. $y=1+4=5$.

Критерии оценки.

На полный балл (12 баллов) 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)

- Мои четыре внучки - замечательные девочки, - рассказывала бабушка Пелагея с нескрываемой гордостью. Каждая из них играет на каком-нибудь музыкальном инструменте и говорит на одном из иностранных языков.

- На чем играет Маша? - спросил я.

- На рояле.

- А кто играет на скрипке?

- Помню только что это та девочка, которая говорит по-французски - ответила бабушка.

Поговорив с бабушкой, я также узнал, что Оля играет на виолончели, а Лена не говорит по-немецки. Маша не знает итальянского языка, а Оля не владеет английским. Валя не знает французского, Лена не играет на арфе, а виолончелистка не говорит по-итальянски. Я совсем запутался. Скажите, кто на каком инструменте играет, и на каком языке говорит?

Критерии:

- -5 баллов за каждое несоответствие

Решение:

1. Так как на арфе играет не Маша, не Оля, и не Лена, значит на арфе играет Валя, а Лена играет на скрипке.
2. Т.к. Лена играет на скрипке, то она говорит по-французски.
3. Оля не говорит по-английски. Она играет на виолончели, следовательно не говорит по-итальянски. Она также не говорит по-французски, т.к. по-французски говорит Лена. Значит Оля говорит по-немецки.
4. Т.к. Маша не говорит по-итальянски, по-французски и по-немецки, она говорит по-английски, а Валя по-немецки.

	Рояль	Скрипка	Арфа	Виолончель	Фр.	Нем.	Ит.	Англ.
Маша	+	-	-	-	-	-	-	+
Оля	-	-	-	+	-	+	-	-
Лена	-	+	-	-	+	-	-	-
Валя	-	-	+	-	-	-	+	-

Задание 3 (12 баллов)

Дед Мороз раздал подарки на Новый год 30 детям. 15 ребят рассказали стишок. 10 ребят спели песенку. 7 ребят спели песенку и сплясали. 5 ребят рассказали стишок и сплясали. 3 ребёнка пели, и плясали, и декламировали стишок. Сколько ребят сплясало, если известно, что просто так подарок не дали никому?

Ответ: 14

Решение.

Всего пели песенку 10 человек. Известно, что 7 при этом сплясали, 3 и пели, и плясали, и рассказывали стишок. Отсюда следует, что детей, которые бы пели песенку и при этом не плясали, нет. Значит, подставим по формуле Байеса числа.

$$X+15+10-7-5+3=30. X+16=30, X=14.$$

Критерии. На полный балл 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 автомата подаётся четырёхзначное число в восьмеричной системе счисления. Автомат выполняет следующие действия:

- переводит число в двоичную систему счисления;
- все разряды числа (двоичного) переписывает в обратном порядке;
- переводит число в шестнадцатеричную систему счисления.

Укажите максимальное число (в восьмеричной системе счисления), такое, что подав его на вход автомата, мы получим трёхзначное число (в восьмеричной системе) у которого все цифры чётные.

Решение

т.к. надо получить 3-хзначное число то младшая цифра исходного числа должны быть равна 0 остальные должны дать 0 в младшем разряде триады после поворота т.е. 3330

Критерии (макс. 12 баллов)

- -6 если ответ не в той системе счисления
- -6 если ответ 3333
- -8 если ответ 2220
- -10 если ответ 1110
- -4 нет решения

Задание 5 (12 баллов)

В кабине лифта 20-этажного дома есть 2 кнопки. При нажатии на одну из них лифт поднимается на 13 этажей, при нажатии на другую — опускается на 8 этажей. Как попасть с 13-го этажа на 8-й?

Решение

№ пп	Начальное состояние (номер этажа)	Нажатие кнопки	Конечное состояние (номер этажа)
1	13	-8	5
2	5	+13	18
3	18	-8	10
4	10	-8	2
5	2	+13	15
6	15	-8	7
7	7	+13	20
8	20	-8	12
9	12	-8	4
10	4	+13	17
11	17	-8	9
12	9	-8	1

13	1	+13	14
14	14	-8	6
15	6	+13	19
16	19	-8	11
17	11	-8	3
18	3	+13	16
19	16	-8	8

Критерии

Оценивается только решение, приведшее к правильному результату.

Если количество нажатий на кнопку превышает количество нажатий на кнопку эталонного решения, то вычитается 3 балла за каждое лишнее нажатие на кнопку.

Задание 6 (15 баллов)

Классный руководитель хочет посмотреть, насколько соотносятся оценки выпускников школы с результатами сдачи ЕГЭ по информатике. Для этого ему нужно знать средний балл двоечников, троечников, хорошистов и отличников. Однако электронный журнал находится на профилактике, и поэтому он получает данные через соцсеть. Требуется найти средний балл каждой категории учащихся.

Формат входных данных:

На первой строке задаётся целое число N - количество ответов на запрос классного руководителя

В следующих N строках записан ответ учащегося. Ответ гарантированно содержит ровно два целых неотрицательных числа, ограниченных пробелом или концом строки. Первое число является оценкой школьника по предмету, второе - количеством баллов по ЕГЭ.

Вам необходимо определить средние баллы каждой из четырёх категорий учащихся. Если учащихся такой категории среди ответов нет, средний балл равен 0.

Пример:

Входные данные:

4

Оценка 5, баллов 86.

Оценка 4, баллов 75

4 и 73

Оценка 3, а вот баллов 76

Выходные данные:

0 76 74 86

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.

//по очереди читаем строки

//извлекаем из каждой строки оценку и балл, после чего

//считаем средние арифметические

```
program league;
const M=5;
type tmas=array [2..M] of integer; //массив, индексы которого совпадают с оценками
procedure cut(s:string; var x,y:integer);
var s1:string;
    flag1,flag2:boolean;
    code:integer;
begin
s:=s+' ';
while pos(' ',s)>0 do
    delete(s,pos(' ',s),1); //чистим лишние пробелы
    flag1:=true; //истина, пока не взято 1 число
    flag2:=true; //истина, пока не взято 2 число
    while flag1 do //пока не взято первое число
        begin
s1:=copy(s,1,pos(' ',s)-1); //извлекаем слово
delete(s,1,pos(' ',s));
val(s1,x,code); //пытаемся преобразовать слово в оценку
if code=0 then //если результат удачный, то опускаем флаг
            begin
                flag1:=false;
            end;
        end;
    while flag2 do
```

```

begin
  s1:=copy(s,1,pos(' ',s)-1);
  delete(s,1,pos(' ',s));
  val(s1,y,code); //пытаемся преобразовать слово в средний балл
  if code=0 then //если результат удачный, то опускаем флаг
    begin
      flag2:=false;
    end;
  end;
end;

```

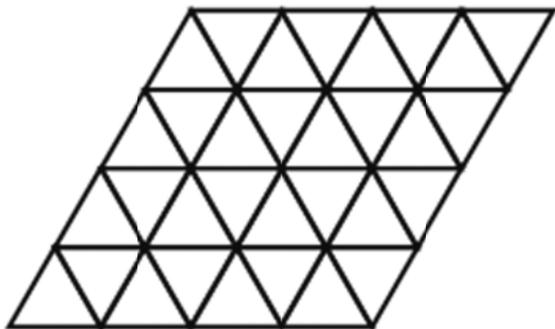
```

var n,x,y,i:integer;
    ball,kolvo:tmas;
    s,name:string;
begin
  readln(n); //читаем количество строк
  for i:=2 to M do //инициализация переменных
    begin
      ball[i]:=0;
      kolvo[i]:=0;
    end;
  for i:=1 to n do
    begin
      readln(S); //читаем очередную строчку
      cut(S,x,y); //извлекаем два числа - оценку и баллы по ЕГЭ
      ball[x]:=ball[x]+y; //добавляем их в массивы
      kolvo[x]:=kolvo[x]+1;
    end;
  for i:=2 to M do
    begin
      if kolvo[i]<>0 then
        write(ball[i]/kolvo[i],' ') //выводим среднее
      else
        write(0,' ');
      end;
    end.

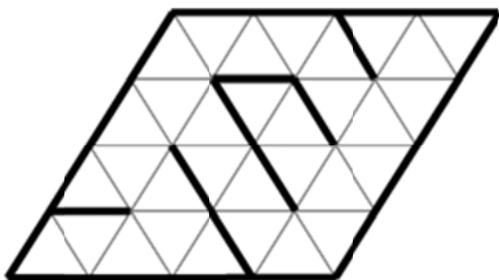
```

Задание 7 (25 баллов)

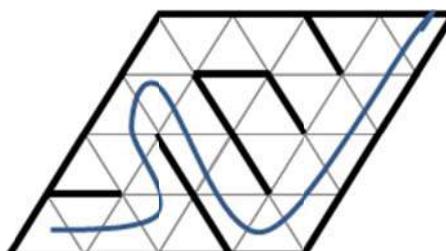
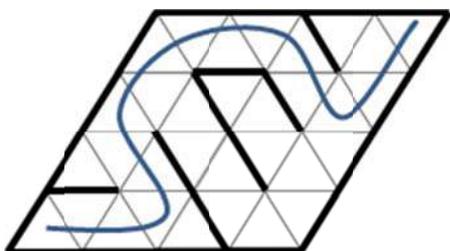
Некоторый замок состоит только из комнат, которые представляют собой равносторонние треугольники (см рис.1)



Не в каждой из стен комнат есть двери. Обозначим на рисунке стены, в которых есть двери тонкими линиями, а в которых нет дверей толстыми.



Посетителю замка необходимо добраться из левого нижнего угла в правый верхний, посетив наименьшее количество комнат.



Выше приведены рисунки для двух способов перемещения в конечную комнату. В первом случае путь включает в себя 18 комнат, а во втором 20 (начальная и конечная комната учитываются).

Необходима написать программу, которая для заданного замка находит кратчайший путь и выводит количество комнат, которые необходимо посетить посетителю замка.

Исходные данные берутся из файла. В первой строке файла через пробел записываются количество рядов комнат N и количество комнат в ряду M .

В следующих $2*N+1$ строках описываются стены комнат – стена с дверью обозначается цифрой 0, без двери – цифрой 1. Для замка, приведённого на рисунке, файл будет иметь следующий вид.

4 8

1111

100001001

0100

100101001

0000

100101001

1000

100100001

1111

Ответ

18

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 (стены замка, наличие или отсутствие дверей), 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

Каждая комната имеет три стены. Будем для каждой стены записывать

координаты связанной с ней комнаты, если в стене есть дверь
и записывать нули, если двери нет (то есть через данную стену комната не связана с другими)
Затем запускаем классическую волну, учитывая, что соседними комнатами для текущей
комнаты будут те, с которыми указана связь (то есть не нули)

}

const

row = 10;

col = 20;

type

coord = **record**

rw, cl: **integer**;

end;

link = **array**[1..3] **of** coord;

pole = **record**

inf: **integer**;

p: link;

end;

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

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

var

i, j, k: **integer**;

begin

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

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

with zamok[i, j] **do**

begin

inf := 0;

for k := 1 **to** 3 **do**

begin

p[k].rw := 0; p[k].cl := 0;

end

end;

end;

procedure razb0(**var** zamok: mas; k: **integer**; s: **string**);

```

var
  i:integer;
begin
  for i := 1 to length(s) do
    if copy(s, i, 1) = '0' then //если в текущей стене дверь, то
      begin
        замок[k div 2, 2 * i - 1].p[1].rw := k div 2 + 1;
        замок[k div 2, 2 * i - 1].p[1].cl := 2 * i;
        замок[k div 2 + 1, 2 * i].p[1].rw := k div 2;
        замок[k div 2 + 1, 2 * i].p[1].cl := 2 * i - 1;
      end;
    end;
  procedure razb1(var замок: mas; k: integer; s: string);
  var
    i:integer;
  begin
    for i := 1 to length(s) do
      if copy(s, i, 1) = '0' then //если в текущей стене дверь, то
        begin
          замок[k div 2 + 1, i-1].p[2].rw := k div 2 + 1;
          замок[k div 2 + 1, i-1].p[2].cl := i;
          замок[k div 2 + 1, i ].p[3].rw := k div 2 + 1;
          замок[k div 2 + 1, i ].p[3].cl := i-1;
        end;
      end;
    procedure inp(var f: text; var замок: mas; n: integer);
  var
    i: integer;
    s: string;
  begin
    readln(f); //первую строчку пропускаем (внешняя стена замка - дверей нет)
    for i := 1 to 2 * n - 1 do //перебираем все остальные строчки, кроме последней
      begin
        readln(f, s); //считываем строку с дверьми
        if i mod 2 = 0 then //если строка четная, то
          razb0(замок, i, s) //вносим наличие дверей в горизонтальных (по рисунку) стенках
        else
          razb1(замок, i, s); //вносим наличие дверей в вертикальных (по рисунку) стенках
      end;
    end;
  end;
end;

```

```

end;
end;
procedure volna(zamok:mas; n, m:integer;var res:integer);// запускаем волну
var
i,j,k:integer;
begin
zamok[n,1].inf:=1; //ставим 1 в начало пути
res:=1;
repeat //повторяем
for i:=1 to n do //перебираем все комнаты
for j:=1 to m do
with zamok[i,j] do
begin
if inf=res then //если в текущую комнату попали на предыдущем ходе ,то
for k:=1 to 3 do //перебираем стены
begin
if p[k].rw<>0 then //если есть дверь, то
if zamok[p[k].rw,p[k].cl].inf=0 then //если еще не были в соседней комнате
zamok[p[k].rw,p[k].cl].inf:=res+1; //делаем шаг в соседнюю комнату
end;
end;
end;
inc(res);
until zamok[1,m].inf<>0 //пока не дойдем до конечной точки
end;
var
f: Text;
n, m, res: integer;
zamok: mas;
begin
Assign(f, 'input.txt');
Reset(f);
init(zamok); //обнуляем
readln(f, n, m);// считываем размеры
inp(f, zamok, n);// считываем конфигурацию
close(f);
volna(zamok, n, m, res);// запускаем волну
writeln(res);// выводим результат
end.

```

Второй (заключительный) этап академического соревнования

Олимпиады школьников «Шаг в будущее»

по общеобразовательному предмету «Информатика»

10 класс, февраль, 2016 г.

Вариант № 3.

Задание 1 (12 баллов)

Определить основание системы счисления, в которой записано выражение:

$$\begin{array}{r} aba_y \\ + b4_y \\ \hline b00_y \end{array}$$

где a и b - цифры числа.

Ответ: 5

Решение. Исходя из формулы, запишем систему уравнений.

$$a+4=y$$

$$b+b+1=y$$

$$a+1=b$$

Очевидно, что $a+4=2a+3$, следовательно $a=1$. $y=1+4=5$.

Критерии оценки.

На полный балл (12 баллов) 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)

- Мои четыре внучки - замечательные девочки, - рассказывала бабушка Пелагея с нескрываемой гордостью. Каждая из них играет на каком-нибудь музыкальном инструменте и говорит на одном из иностранных языков.

- На чем играет Маша? - спросил я.

- На рояле.

- А кто играет на скрипке?

- Помню только что это та девочка, которая говорит по-французски - ответила бабушка.

Поговорив с бабушкой, я также узнал, что Оля играет на виолончели, а Лена не говорит по-немецки. Маша не знает итальянского языка, а Оля не владеет английским. Валя не знает

французского, Лена не играет на арфе, а виолончелистка не говорит по-итальянски. Я совсем запутался. Скажите, кто на каком инструменте играет, и на каком языке говорит?

Критерии:

- -5 баллов за каждое несоответствие

Решение:

1. Так как на арфе играет не Маша, не Оля, и не Лена, значит на арфе играет Валя, а Лена играет на скрипке.
2. Т.к. Лена играет на скрипке, то она говорит по-французски.
3. Оля не говорит по-английски. Она играет на виолончели, следовательно не говорит по-итальянски. Она также не говорит по-французски, т.к. по-французски говорит Лена. Значит Оля говорит по-немецки.
4. Т.к. Маша не говорит по-итальянски, по-французски и по-немецки, она говорит по-английски, а Валя по-немецки.

	Рояль	Скрипка	Арфа	Виолончель	Фр.	Нем.	Ит.	Англ.
Маша	+	-	-	-	-	-	-	+
Оля	-	-	-	+	-	+	-	-
Лена	-	+	-	-	+	-	-	-
Валя	-	-	+	-	-	-	+	-

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 сдаёт 0 человек. Следовательно, пересечение трёх 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)

Классный руководитель хочет посмотреть, насколько соотносятся оценки выпускников школы с результатами сдачи ЕГЭ по информатике. Для этого ему нужно знать средний балл двоечников, троечников, хорошистов и отличников. Однако электронный журнал находится на профилактике, и поэтому он получает данные через социальную сеть. Требуется найти средний балл каждой категории учащихся.

Формат входных данных:

На первой строке задаётся целое число N - количество ответов на запрос классного руководителя

В следующих N строках записан ответ учащегося. Ответ гарантированно содержит ровно два целых неотрицательных числа, ограниченных пробелом или концом строки. Первое число является оценкой школьника по предмету, второе - количеством баллов по ЕГЭ.

Вам необходимо определить средние баллы каждой из четырёх категорий учащихся. Если учащихся такой категории среди ответов нет, средний балл равен 0.

Пример:

Входные данные:

4

Оценка 5, баллов 86.

Оценка 4, баллов 75

4 и 73

Оценка 3, а вот баллов 76

Выходные данные:

0 76 74 86

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.

//по очереди читаем строки

//извлекаем из каждой строки оценку и балл, после чего

//считаем средние арифметические

```
program ege;
```

```
const M=5;
```

```
type tmas=array[2..M] of integer; //массив, индексы которого совпадают с оценками
```

```
procedure cut(s:string; var x,y:integer);
```

```
var s1:string;
```

```
    flag1,flag2:boolean;
```

```
    code:integer;
```

```
begin
```

```
  s:=s+' ';
```

```
  while pos(' ',s)>0 do
```

```
    delete(s,pos(' ',s),1); //чистим лишние пробелы
```

```
    flag1:=true; //истина, пока не взято 1 число
```

```
    flag2:=true; //истина, пока не взято 2 число
```

```
    while flag1 do //пока не взято первое число
```

```
      begin
```

```
        s1:=copy(s,1,pos(' ',s)-1); //извлекаем слово
```

```
        delete(s,1,pos(' ',s));
```

```
        val(s1,x,code); //пытаемся преобразовать слово в оценку
```

```
        if code=0 then //если результат удачный, то опускаем флаг
```

```
          begin
```

```
            flag1:=false;
```

```
          end;
```

```
        end;
```

```
    while flag2 do
```

```
      begin
```

```

s1:=copy(s,1,pos(' ',s)-1);
delete(s,1,pos(' ',s));
val(s1,y,code); //пытаемся преобразовать слово в средний балл
if code=0 then //если результат удачный, то опускаем флаг
  begin
    flag2:=false;
  end;
end;
end;
var n,x,y,i:integer;
    ball,kolvo:tmas;
    s,name:string;
begin
  readln(n); //читаем количество строк
  for i:=2 to M do //инициализация переменных
  begin
    ball[i]:=0;
    kolvo[i]:=0;
  end;
  for i:=1 to n do
  begin
    readln(S); //читаем очередную строчку
    cut(S,x,y); //извлекаем два числа - оценку и баллы по ЕГЭ
    ball[x]:=ball[x]+y; //добавляем их в массивы
    kolvo[x]:=kolvo[x]+1;
  end;
  for i:=2 to M do
  begin
    if kolvo[i]<>0 then
      write(ball[i]/kolvo[i],' ') //выводим среднее
    else
      write(0,' ');
    end;
  end;
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.

63

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
    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
sosedi(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');
Reset(f);
init(soty); //обнуляем
readln(f, n, m);// считываем размеры
inp(f, soty, n,m);// считываем конфигурацию
close(f);
volna(soty, n, m, res);// запускаем волну
writeln(res);// выводим результат
end.

```

**Второй (заключительный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее»
по общеобразовательному предмету «Информатика»
10 класс, март, 2016 г.**

Задание 1 (12 баллов)

Определить минимальное основание системы счисления y , для которого существует решение уравнения

$$3_y x_y + 5_y x_y = 20_y. X - \text{целое число.}$$

Ответ: 8

Решение. Перепишем уравнение в виде $8x=2y$. Очевидно, что $4x=y$, следовательно, основание системы счисления кратно 4. 4 нам не подходит, так как в троичной системе счисления нет цифры 5. Следовательно, ответ 8.

Критерии оценки.

На полный балл (12 баллов в 9 и 10 классах) 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)

Из Москвы в Санкт-Петербург едут Сидоров, Иванов и Петров. Фамилии этих пассажиров распространённые, так что оказалось, что такие же фамилии у трёх человек из поездной бригады – машиниста, помощника машиниста и проводника.

Известно, что все:

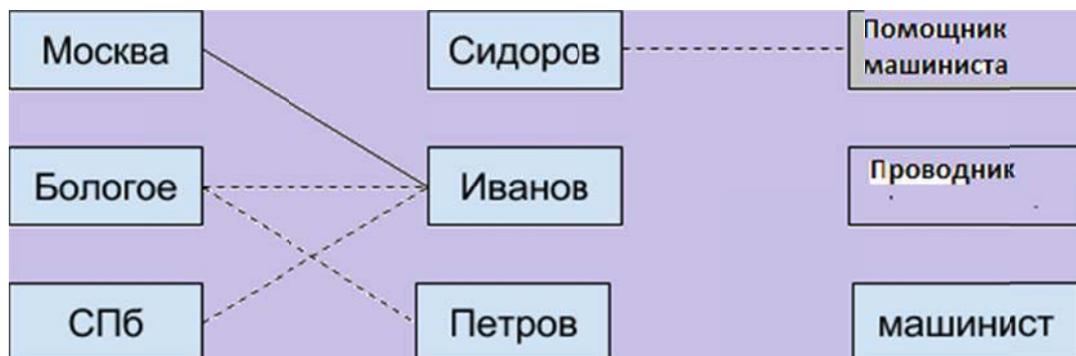
1. перечисленные лица живут в Москве, Санкт-Петербурге и Бологом;
2. все три пассажира живут в разных городах по Октябрьской железной дороге;
3. все члены поездной бригады живут в одном городе;
4. пассажир Иванов живёт в Москве;
5. проводник живёт в Бологом;
6. пассажир - однофамилец проводника живёт в Санкт-Петербурге;
7. пассажир Петров не проживает там же, где его однофамилец;
8. Сидоров из поездной бригады выиграл у помощника машиниста партию в бильярд.

Каковы фамилии членов поездной бригады и где живут они и пассажиры?

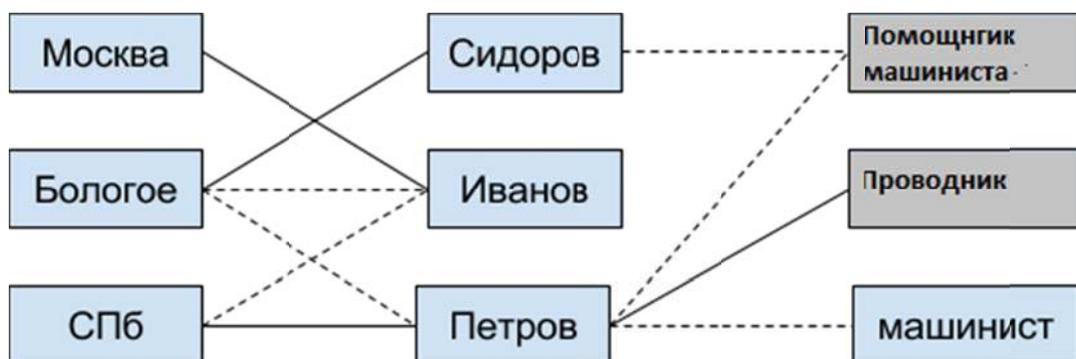
Ответ обоснуйте.

Решение

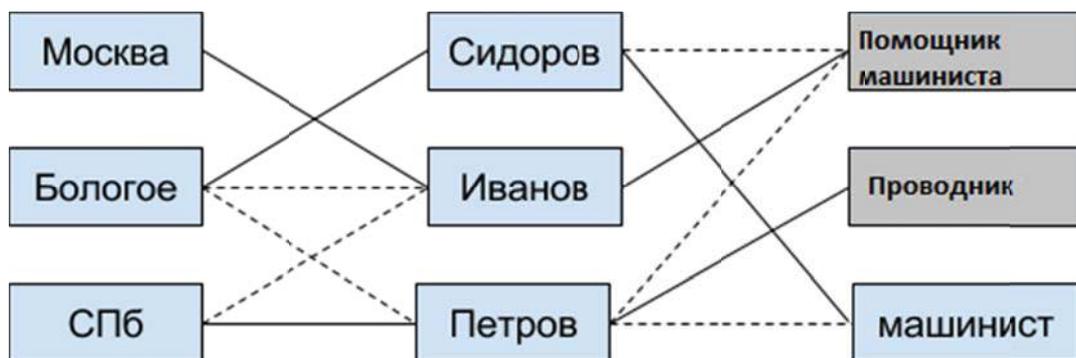
1. Все члены поездной бригады живут в Бологом (п. 3, 5). Пассажир Иванов живет в Москве (п. 4). Из п.7 и того, что вся поездная бригада живет в Бологом, следует, что Петров не живет в Бологом. Из п.8 следует, что Сидоров не помощник машиниста.



2. Из схемы видно, что в Бологом живет Сидоров, значит Петров. Живет в Санкт-Петербурге. Из п.6 и из того, что Петров живет Санкт-Петербурге следует, что фамилия проводника Петров.



3. Из схемы видно, что помощник машиниста -- Иванов, следовательно, Сидоров -- Машинист.



Ответ: Сидоров - Бологое, Иванов - Москва, Петров - СПб, поездная бригада - Бологое.

Сидоров - машинист, Иванов – помощник машиниста, Петров - проводник

Критерии 12 баллов:

- -3 балла за каждое несоответствие

Задание 3 (12 баллов)

В группе студентов 25 человек. К указанной дате домашнее задание по общей физике сделали 10 человек. Домашнее задание по математическому анализу - 8 человек. Из них 6

человек сделали задание и по физике, и по программированию, но не по математическому анализу. 7 человек сделали задание и по математическому анализу, и по программированию, но не по физике. Одновременно математический анализ и физику, но не программирование, не делал никто. Сколько человек выполнили задание строго по одному предмету, если известно, что тех, кто сделал все задание, в группе нет, а всех разгильдяев, которые ничего не делают, отчислили ещё в прошлом семестре, за исключением одного человека?

Ответ: 11

Решение.

Всего хоть что-то делали 24 человека. 10 делали физику, матан - 8. Значит, только программирование сделали $24 - 10 - 8 = 6$ человек. Известно, что 6 человек из 10 делали кроме физики еще программирование. Значит, 4 делали только физику. По аналогии, только матан сделал 1 человек.

$$6 + 4 + 1 = 11.$$

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 2 балла.

Задание 4 (12 баллов)

Автомат получает трёхзначное число, записанное в шестнадцатеричной системе счисления. В этом числе он находит сумму первой и второй и сумму второй и третьей цифр. Обе суммы переводятся в восьмеричную систему счисления, после чего умножаются на 2. И записываются друг за другом в порядке возрастания.

Например, из числа 123_{16} автомат получил число 612_8 .

Найти максимальное число, обработав которое автомат получит двузначное восьмеричное число. Ответ запишите в шестнадцатеричной системе счисления.

Ответ обоснуйте.

Ответ 303.

Решение:

Т.к 8-е число мы получаем умножением цифр на два, то макс. число может быть 110 110. Оно было получено из числа 011 011. Т.е. сумма цифр равна 3. Макс. число суммы первой+второй цифры и второй+третьей цифры которого равны 3 - 303.

Критерии (12 баллов)

- -4 если ответ не в той системе счисления
- -8 в результате обработки получаем двузначное число, но выбран не максимальный вариант.
- -6 нет решения

Задание 5 (22 балла)

Иван Петрович живёт в городе «Дорожный» и каждый день он добирается на работу на машине по дорогам города, на которых бывают пробки. Необходимо выбрать для Ивана Петровича наиболее быстрый маршрут, с помощью которого он сможет добраться до работы.

Исходные данные задаются следующим образом:

- на первой строке пишется количество перекрёстков в городе (натуральное число не превосходящее 1000)
- на второй строке номер перекрёстка, с которого Иван Петрович начинает движение (перекрёстки в городе нумеруются натуральными числами)
- на третьей строке номер перекрёстка, куда должен добраться Иван Петрович
- на четвертой строке далее записывается количество дорог в городе
- на пятой и последующих строках идут данные по каждой дороге в виде трёх натуральных чисел, записанных через пробел: первые два числа - номера перекрёстков, между которыми проходит дорога (по каждой дороге можно ездить в обоих направлениях), третье число - время, необходимое на проезд в минутах.

В качестве ответа выведите через пробел номера перекрёстков, которые надо проехать.

Входные данные	Выходные данные
7 1 7 10 1 2 3 1 3 10 2 4 2 2 5 6 3 4 10 3 6 11 4 5 3	1 2 4 5 7

4 6 10	
5 7 1	
6 7 15	

=====

{Реализация алгоритма Флойда-Уоршелла}

const

nmax = 1000;

INFINITY = MaxInt;

type

TRow = array [0..nmax - 1] of integer;

TVertex = array [0..nmax - 1] of TRow;

procedure FloydWarshall(v: TVertex; n: integer; var d, p: TVertex);

var

i, j, k: integer;

begin

d := v;

for i := 0 to pred(n) do

for j := 0 to pred(n) do

begin

if d[i, j] = 0 then

d[i, j] := INFINITY;

if i = j then

d[i, j] := 0;

p[i, j] := j;

end;

for k := 0 to pred(n) do

begin

for i := 0 to pred(n) do

begin

for j := 0 to pred(n) do

begin

if (d[i, k] <> INFINITY) and (d[k, j] <> INFINITY) then

begin

if (d[i, j] > d[i, k] + d[k, j]) then


```
read(m);
for i:=1 to n do
  for j:= 1 to n do
    Vertex[i,j]:=0;
for i:= 1 to m do
  begin
    read(x,y,l);
    Vertex[x-1,y-1]:=l;
    Vertex[y-1,x-1]:=l;
  end;
AlgoFW(Vertex, n,a-1,b-1);
end.
```

Критерии (макс. 22 баллов):

- -2 балл за каждые 5 синтаксических ошибок
- -4 балла если нет вывода результата
- -3 балла если вывод результата в обратном порядке
- -9 не эффективно по времени / памяти (например, рекурсивный перебор путей)
- -12 если идет поиск самого длинного пути
- -4 за каждую ошибку в алгоритме (например, не зануляем матрице смежности, вместо > написано <)
- -9 за серьезную ошибку в алгоритме (считаем дороги односторонними, например)