Первый (заочный) этап академического соревнования

Олимпиады школьников «Шаг в будущее» по общеобразовательному предмету «Информатика», осень 2016 г.

10 КЛАСС

Вариант 1К

Задание 1 (12 баллов)

Определить основание системы счисления, в которой записано выражение:

$$\begin{array}{l} aba_y \\ \underline{+\ b4_y} \\ b00_y \end{array}$$

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)

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).
- Андрей накануне званого вечера был в гостях у Александра.

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

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, Алексадр - 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	-	-	-	-	+	-	+	-

Критерии оценки:

Обоснование может быть и другим.

Верный ответ + обоснование: 12

Верный ответ без обоснования: 6

обоснование без вывода: 3

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 оценки:

Полный балл ставится за полностью записанное решение с правильным ответом. В целом правильный ответ, содержащий арифметическую ошибку, оценивается в 9 баллов. Только правильный ответ оценивается в 3 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 12316 автомат получил число 6128.

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 оценки (макс. 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 (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 оценки:

Полный балл ставится за полностью записанное решение с правильным ответом. Фрагмент правильного ответа оценивается в 3 балла.

Задание 6 (15 баллов)

На вход программе подаётся последовательность целых чисел (по одному элементу в строке) в диапазоне от -30000 до 30000, количество чисел заранее неизвестно, но не превосходит 1000. Признаком окончания последовательности является «0». Найти среднее арифметическое наиболее длинной подпоследовательности состоящих только из неубывающих нечётных чисел и вывести его с фиксированной точностью два знаков в дробной части. Если есть несколько подпоследовательностей одинаковой (максимальной) длины, то выбрать последнюю из них. Гарантируется, что в последовательности есть хотя бы одно нечётное число.

Решить задачу используя наиболее эффективный по памяти и времени работы алгоритм.

Входные данные	Выходные данные
3 6 -2 -1 -5	3.67

7 9 40 1 3 8 0	

Решение:

```
============================================================
var
 sum, kol, maxkol, ch, pch: integer;
 res: real;
begin
 maxkol := 0;
 read(pch);
 sum := (abs(pch) mod 2) * pch;
 kol := abs(pch) mod 2;
 read(ch);
 while ch <> 0 do
  begin
    if (kol > 0) and (ch mod 2 <> 0) and (ch >= pch) then
      begin
        sum := sum + ch;
        kol := kol + 1;
        if kol > maxkol then
          begin
            maxkol := kol;
            res := sum / kol;
          end;
      end
    else
      begin
        sum := (abs(ch) mod 2) * ch;
```

```
        kol := abs(ch) mod 2;
      end;
    pch := ch;
    read(ch);
  end;
 writeln(res:0:2);
end.
```

==========

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

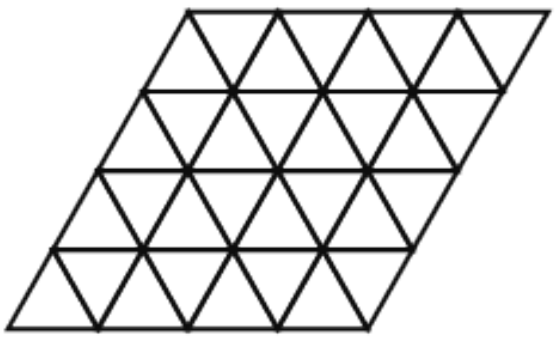
Не оптимальное по расходу памяти решение (например, записать все в массив и найти 4 максимума) оценивается в 11 баллов.

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

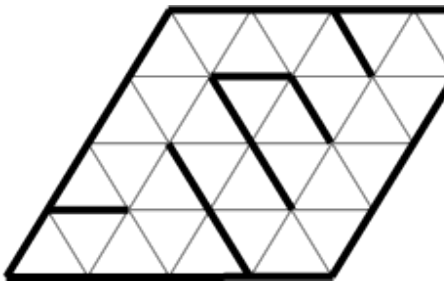
Неверно работающая программа, из которой, тем не менее, понятно, что участник понимает решение задачи, оценивается в 3 балла.

Задание 7 (25 баллов)

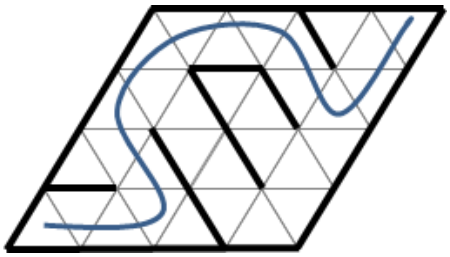
Некоторый замок состоит только из комнат, которые представляют собой равносторонние треугольники (см рис.1)

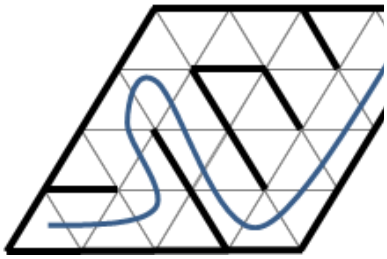


Не в каждой из стен комнат есть двери. Обозначим на рисунке стены, в которых есть двери тонкими линиями, а в которых нет дверей толстыми.



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*M+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

100001001

1111

Ответ

18

```
{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
      zamok[k div 2, 2 * i - 1].p[1].rw := k div 2 + 1;
      zamok[k div 2, 2 * i - 1].p[1].cl := 2 * i;
      zamok[k div 2 + 1, 2 * i].p[1].rw := k div 2;
      zamok[k div 2 + 1, 2 * i].p[1].cl := 2 * i - 1;
    end;
end;

procedure razb1(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
      zamok[k div 2 + 1, i-1].p[2].rw := k div 2 + 1;
      zamok[k div 2 + 1, i-1].p[2].cl := i ;
      zamok[k div 2 + 1, i ].p[3].rw := k div 2 + 1;
      zamok[k div 2 + 1, i ].p[3].cl :=  i-1;
    end;
end;

procedure inp(var f: text; var zamok: 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(zamok, i, s) //заносим наличие дверей в горизонтальных (по рисунку) стенках
    else
      razb1(zamok, i, s); //заносим наличие дверей в вертикальных (по рисунку) стенках
  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;
       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.
```

Критерии оценки

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.