

Разбор задач муниципального этапа Всероссийской олимпиады школьников для 10-11 классов 2017-18 учебного года

Задача 1. Древняя Курская земля

Основная проблема определить пересекаются ли участки раскопки, если не пересекаются, то площадь очевидна $w1*h1+w2*h2$, но если участки пересекаются, то площадь раскопок уменьшается на площадь пересечения

Возможное решение:

```
var X1, Y1, W1, H1, X2, Y2, W2, H2, x3, x4, y3, y4, d, v, a, b, c1, c2, s1:
integer;
begin
  readln(X1, Y1, W1, H1);
  readln(X2, Y2, W2, H2);
  x3:=X1+W1; X4:=X2+W2;
  Y3:=Y1+H1; Y4:=Y2+H2;
  {Считаем, что первый участок (с координатами X1,Y1) имеет меньшие координаты
  левого нижнего угла по оси Y чем второй, если это не так, то, для уменьшения
  проверок, меняем участки местами}
  if y2<=y1 then begin c1:=x1;c2:=y1;x1:=x2; y1:=y2; x2:=c1;
    y2:=c2; c1:=x3;c2:=y3;x3:=x4; y3:=y4;
    x4:=c1; y4:=c2; end;
  if (y2>=y3) or (x4<=x1) or (x3<=x2) then s1:=0 {участки не
  пересекаются}
  else begin {участки пересекаются}
    if x2<x1 then a:=x1 else a:=x2;
    if x4<x3 then b:=x4 else b:=x3;
    d:=b-a;
    if y4>y3 then a:=y3
    else a:=y4;
    v:=a-y2;
    s1:=d*v; {площадь пересечения}
  end;
  writeln(w1*h1+w2*h2-s1);
end.
```

Задача 2. English ЕГЭ

Для подсчета слов используем следующее свойство: перед первой буквой каждого слова стоит не буква. Исключение составляет буква, слева от которой стоит символ “-“, а перед ним - буква. Осталось написать программу, которая перебирает все символы строки, и считает, сколько среди них первых букв.

Для того, чтобы не было выхода за пределы строки в начале работы, перед первым символом строки добавляется два символа-разделителя (в программе это «!!»)

```

readln(s);
s:='!!'+s;
for i:=3 to length(s) do
if (s[i] in ['a'..'z', 'A'..'Z']) //s[i] - это буква
    and not (s[i-1] in ['a'..'z', 'A'..'Z']) //s[i-1] - это не буква
    and not ((s[i-1]='-') and (s[i-2] in ['a'..'z', 'A'..'Z']))
//s[i-1] - это не дефис
    then n:=n + 1;
writeln(n);

```

Задача 3. Игровой код

Один из алгоритмов решения задачи – это моделирование описанного в задаче процесса: переводим исходное число в двоичную систему счисления, получаем все его циклические сдвиги, переводим их в десятичную систему и ищем максимум:

```

Var I, J, Max, N, ND, D: Integer;
    Digits: array[1..20] Of Integer;
Begin
    assign(input, 'gamecod.in');
    reset(input);
    assign(output, 'gamecod.out');
    rewrite(output);
    readln(N);
    Max:= N;
    ND := 0;
//---перевод в двоичную систему-----
    While N>0 Do
        Begin
            Inc(ND);
            Digits[ND] := N Mod 2;
            N := N Div 2
        End;
//---моделирование циклических сдвигов-----
    For I:=1 To ND-1 Do
        Begin //---сдвиг---
            D:=Digits[1];

```

```

For J:=2 To ND Do Digits[J-1] := Digits[J];
Digits[ND]:=D;
//---перевод в десятичную систему-----
N:=0;
For J:=ND DownTo 1 Do N:=N*2+Digits[J];
//---сравнение с максимумом-----
If N>Max Then Max:=N
End;
writeln(Max);
close(output);
close(input);
End.

```

Можно применить операцию побитового сдвига SHR, не переводя числа явно в двоичную систему.

Задача 4. Приз

Можно использовать такой алгоритм. Завести две переменные для хранения текущих координат участника игры. Координаты будут изменяться соответственно входным данным. У нас имеется 8 направлений, т.е. 8 случаев, в каждом из которых координаты меняются соответствующим образом. Эти 8 случаев-направлений пронумерованы подряд по часовой стрелке. Используя это, можно получить, например, такую реализацию алгоритма:

```

X:=0;
Y:=0;
For I:=1 to N do
Begin
  Readln(D,L);
  X:=X+Sin(Pi/4*(D-1))*L;
  Y:=Y+Cos(Pi/4*(D-1))*L;
End;
Write(X:0:3,' ',Y:0:3);

```