

Разбор задач муниципального этапа олимпиады по информатике

1. Кузнечик (7-8 класс)

Тема: разбор случаев, арифметика

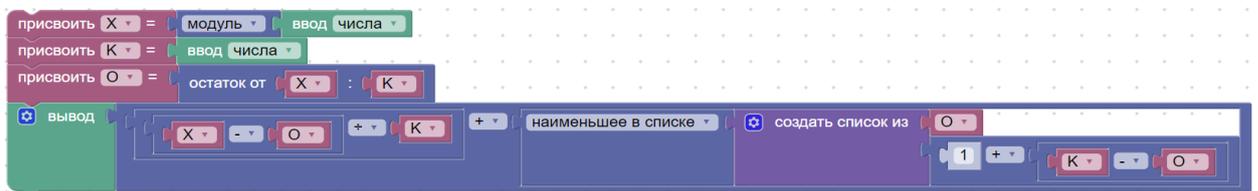
Сложность: простая

В случае отрицательного значения X количество прыжков будет совпадать с $|X|$. Если X делится без остатка на K , то используем только большие прыжки. Если есть остаток $O = X \bmod K > 0$, то его можно добрать маленькими прыжками от координаты кратной K , можно недопрыгнуть до X , и сделать O маленьких прыжков направо, либо перепрыгнуть X дополнительным большим прыжком, а затем вернуться в X сделав $K-O$ маленьких прыжков налево. Выбираем минимум из этих двух вариантов.

Пример реализации:

```
var x, k, o: longint;  
begin  
  read(x, k);  
  if x < 0 then x := -x;  
  o := x mod k;  
  if o > k - o then  
    o := k - o + 1;  
  writeln(x div k + o);  
end.
```

Пример реализации на Blockly:



2. Весовщик (7-8 классы)

Тема: исполнители, разбор случаев

Сложность: простая

Принимаем решение в зависимости от результатов взвешивания. Для решения подзадачи 4 нужно придумать оптимальный порядок действий, перебрав варианты

Пример реализации:



3. Алгоритм

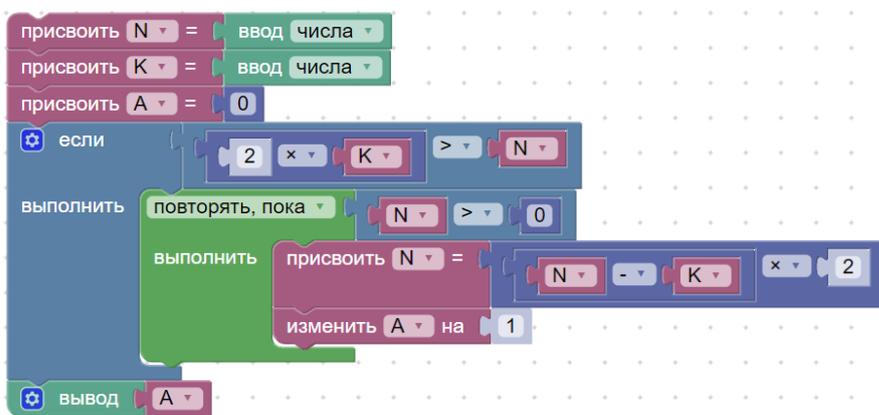
Тема: реализация программы по схеме алгоритма

Сложность: простая

Пример реализации:

```
var n, k, a: integer;
begin
  read(n);
  read(k);
  a:=0;
  if 2*k>n then
  begin
    repeat
      n:=(n-k)*2;
      a:=a+1;
    until n<=0;
  end;
  writeln(a);
end.
```

Пример реализации на Blockly:



4. Идеальная стена

Тема: делители числа, разбор случаев

Сложность: средняя

Подзадача 1: Если все детали одинаковые, то мы можем сложить из них единственную стену с шириной, равной ширине детали, и высотой, равной количеству деталей.

Подзадача 2:

Если количество квадратных деталей четно, то можно собрать стену ширины 2 и высоты $A/2+1$.

Иначе сложить стену невозможно.

Подзадача 3:

Подсчитаем площадь стены S , равную $A+2*B$. Ширина (W) и высота (H) стены должны являться делителями S , поэтому будем перебирать соответствующие пары делителей. Для каждой пары проверим, можно ли сложить стену соответствующего размера, следующим образом:

- мы должны иметь не меньше $W-1$ прямоугольных деталей, так как у нас есть $W-1$ место для вертикального разреза, и каждое из них должно быть перекрыто прямоугольной деталью хотя бы в одном из горизонтальных рядов;

- эти детали могут быть уложены в "шахматном порядке" в пределах двух любых горизонтальных рядов, остальные в этом случае могут выкладываться произвольно, без учета возможных вертикальных разрезов;

- если ширина нечетна, то в каждом горизонтальном ряду должен быть по крайней мере одна квадратная деталь;

- если ширина четна, то требуется по крайней мере 2 квадратных детали в одном из горизонтальных рядов, иначе все прямоугольные детали будут находиться в позициях с одинаковой четностью и не перекроют половину потенциальных разрезов;

- случаи $W=2$ и $H=1$ могут потребовать отдельной обработки, так как в этом случае рассмотренное построение из двух рядов "в шахматном порядке" не применимо.

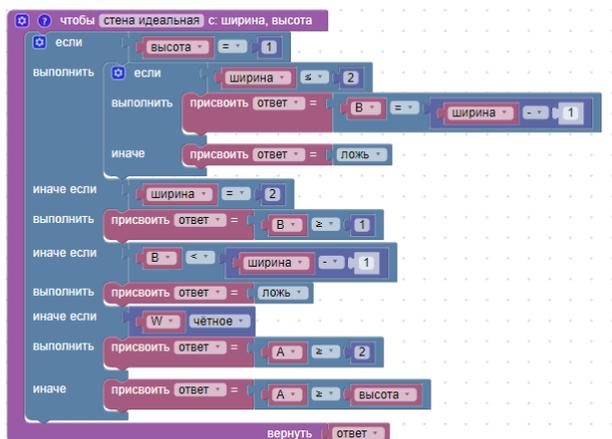
Подзадача 4:

Решается аналогично подзадаче 3, но перебор делителей должен выполняться не более чем за квадратный корень из S .

Пример реализации:

```
var a,b,s,w,h:int64;
    r1,r2:array[1..10000] of int64;
    i,n1,n2:integer;
function check(w,h:int64):boolean;
begin
    if h=1 then { высота стены 1 }
        result:=(w=2) and (b=1) or (w=1) and (b=0)
    else if w=2 then { ширина стены 2 }
        result:=b>0
    else if b<w-1 then { на каждый из w-1 вертикальных разрезов
        должна быть прямоугольная деталь, его перекрывающая }
        result:=false
    { прямоугольных деталей достаточно и горизонтальных рядов по крайней мере 2 }
    else if w mod 2 = 1 then {в каждом горизонтальном
        ряду нужен хотя бы 1 квадратная деталь }
        result:=a >= h
    else { нужны только 2 квадратных детали в одном из рядов }
        result:=a >= 2;
end;
begin
    read(a,b);
    s:=a+2*b;
    w:=1;
    n1:=0;
    n2:=0;
    while w*w<=s do
    begin
        if s mod w=0 then
        begin
            h:=s div w;
            if check(w, h) then
            begin
                inc(n1);
                r1[n1]:=w;
            end;
            if (w<>h) and check(h,w) then
            begin
                inc(n2);
                r2[n2]:=h;
            end;
        end;
        inc(w);
    end;
    writeln(n1+n2);
    for i:=1 to n1 do
        writeln(r1[i], ' ',s div r1[i]);
    for i:=n2 downto 1 do
        writeln(r2[i], ' ',s div r2[i]);
end.
```

Пример реализации на Blockly



```

присвоить A = ввод числа
присвоить B = ввод числа
присвоить S = A * 2 * B
присвоить делители = создать пустой список
цикл по W от 1 до квадратный корень S с шагом 1
  выполнить если S делится на W
    выполнить присвоить H = S / W
    выполнить если стена идеальная с:
      ширина W
      высота H
    выполнить в списке делители вставить в последний W
    выполнить если W^2 ≠ S и стена идеальная с:
      ширина H
      высота W
    выполнить в списке делители вставить в последний H
присвоить делители = сортировать числовая по возрастанию делители
вывод длина делители
для каждого элемента W в списке делители
  выполнить присвоить H = S / W
  выполнить вывод W
  выполнить вывод H

```

5. Чертежник

Тема: исполнители, задача на построение

Сложность: ниже среднего

Нужно придумать способ рисования n-угольника. Это может быть лесенка, расчёска (см. рис) и т. д. Для нечетного n в фигуре должно быть на 1 угол меньше. В подзадаче 1 каждая фигура может иметь свою форму.



Пример реализации:

```

если n чётное
  выполнить присвоить k = n / 2 - 1
  повторить k раз
    выполнить двигаться вперед
    повернуть на -90
    двигаться вперед
    повернуть на -45
  иначе
    присвоить k = n / 2
    повторить k + 1 раз
      выполнить двигаться вперед
      повернуть на -135
      двигаться вперед
    повторить k раз
      выполнить двигаться вперед
      повернуть на -135
      двигаться вперед
      повернуть на 135
    повернуть на -135
    двигаться вперед

```