

Олимпиада школьников «Надежда энергетики»

I10F01 ДИСТАНЦИОННО,  
С ИСПОЛЬЗОВАНИЕМ ВКС

№ группы

Место проведения

QV 97-15

шифр

— Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 43101

ФАМИЛИЯ АЛЕКСАНДИН

ИМЯ АНТОН

ОТЧЕСТВО ПАВЛОВИЧ

Дата рождения 09.08.2004

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 13 листах

Дата выполнения работы: 18.02.2021  
(число, месяц, год)

Подпись участника олимпиады: Александрин

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа и ранее справа

/1

Лист 1

 $A \rightarrow C$  - истина $A \& B \rightarrow D$  - истина $\neg B \rightarrow E$  - истина

C, D, E - лямбда (0)

~~или~~~~Составим таблицу истинности~~

A	B	A → B
0	0	1
0	1	1
1	0	0
1	1	1

1) макс макс  $A$  и  $C=1$   
и  $C=0$ , то  
 $A=0$ , то лямбда  
 $A=0$

2) из утверждения  $\neg B \rightarrow 0 \Rightarrow \neg B=0 \Rightarrow$   
 $\Rightarrow B=1$ , то лямбда истина.

3) Проверим через  $A \& B \Rightarrow D=1$   
 $0 \& 1 \rightarrow 0 = 1$   
ложно

+

ВНИМАНИЕ! Проводятся только те, что записано с этой стороны листа в рамках «справ»

№4 лист 2  
 Для решения данной задачи, нужно:

- 1) заранее задать двумерный массив матрицы  $n \times m$  числами
- 2) организовать проход по массиву от вершинам  $[N, 1]$  до вершинам  $[1, M]$ , при этом на каждом шаге нужно проверить, является ли число в ячейке пустым. Если да, то увеличиваем  $count$  на 1.

Проход по массиву осуществляется с помощью рекурсивной переменной  $trig$ , если она не ноль, то идти по массиву по вершинам вверх, иначе вниз. Если вершина не по пути проверки, вернуть 0, если по пути первого элемента, то вернуть  $count$  по рекурсии.

Алгоритм определяемая функция  $check$  стандартной, цикл от 2 до  $count$  включительно перебираем  $count$ . Если мы не нашли число, то вернуть 0, если нашли, то вернуть  $count$  + 1. (Проход же не  $count$ )



ВНИМАНИЕ! Проверьте только то, что записано с этой стороны листа в рамке справа

(Продолжение №4) лист 3

цел  $N$   
цел  $N$   
цел  $N$

цел  $N$

цел  $N$

цел  $i$  от  $1$  до  $N$  и массив

цел  $i$  от  $1$  до  $N$   
цел  $j$  от  $1$  до  $N$

цел  $i$  от  $1$  до  $N$   
цел  $j$  от  $1$  до  $N$

цел  $i$  от  $1$  до  $N$   
цел  $j$  от  $1$  до  $N$   
цел  $i$  от  $1$  до  $N$   
цел  $j$  от  $1$  до  $N$

цел  $i$  от  $1$  до  $N$

цел  $r = x$

цел  $k = 0$  и  $N$

цел  $y = N$

цел  $y = N$

цел  $tr =$  инициализация

цел  $tr =$  инициализация

(Продолжение листа №4)



ВНИМАНИЕ! Проверьте только то, что записано с этой стороны листа в рамке справа

(Продолжение №4) лист 4

~~к<sub>у</sub>~~ ПОКА (не x=1) и (не y=1)

к<sub>у</sub>

если (trigg = истина) то

ПОКА (не y=1)

к<sub>у</sub>

y = y - 1

~~for i от 2 до round(sqrt(x))~~  
~~for i от 2 до round(sqrt(array[y,x]))~~

к<sub>у</sub>

если (array[y,x] mod i = 0) то

tr = tr + some

else if (let:let, some:some)

все

к<sub>у</sub>

если (tr = истина) то

k = k + 1

все

~~к<sub>у</sub>~~

к<sub>у</sub>

trigg = ~~false~~ true

иначе

(Продолжение №5) лист 5)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

(Заг. №4)

лист №5

начало

ПОКА (не  $y = 1$ )

НУ

$y = y + 1$

~~для  $i$  от 2 до  $bound$  ( $sqrt(x)$ )~~

для  $i$  от 2 до  $bound$  ( $sqrt(array[y, x])$ )

НУ

если  $array[y, x] \bmod i = 0$  то

$tr = i$  ~~и~~  $break$

$break$  и  $break$  ~~и~~  $break$  из цикла

все

КЦ

если ( $tr =$  неинна) то

$k = k + 1$

все

КЦ

$tr =$  неинна

все

(Продолжение лист №6)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Продолжить решение задачи № 6

если (не  $x = 1$ )

$PT = x - 2$

пока (не  $x = PT$ )

$x = y$

$x = x - 1$

для  $i$  от 2 до round( $sqrt(array[y, x])$ )

Нз

если (array  $[y, x]$  mod  $i = 0$ ) то

$tr = floor(x / i)$

вызвать  $i$  блок, вызывая из функции

else

кз

если ( $tr = ~~1~~$  и т.д.) то

$k = k + 1$

else

кз

else

кз // от цикла "пока  $x$  и  $y <= 1$ "

вызвать  $k$

кон

Проверка N





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа и рамке справа

№5 Лист 4

алгоритм:

- 1) вводим матрицу
- 2) сортируем элементы по координате  $x$
- 3) проверяем монотонность убывания по координате  $y$

алг. функция  
матцел  $i, s, p, m \in \mathbb{N}, 2]$ лог  $\Gamma$ ввод  $n$ цел  $i$  от 1 до  $n$ ввод  $m \in [i, 1]$ ввод  $m \in [i, 2]$ 

кц

цел  $j$  от 1 до  $n-1$  // сортировка

нц

~~цел  $j$  от 1 до  $n-1$~~ цел  $j$  от  $i+1$  до  $n$ если  $m \in [i, 1] > m \in [j, 1]$  то $p = m \in [j, 1]$  $m \in [j, 1] = m \in [i, 1]$  $m \in [i, 1] = p$  $p = m \in [j, 2]$  $m \in [j, 2] = m \in [i, 2]$ 

(Продолжить на листе №2)

Лист 07 из 13



ВНИМАНИЕ! Проверьте только то, что написано с этой стороны листа в рамке справа

(Продолж №5) Лист 1/2

$$m[i, 2] = p$$

всё

кз

кз

$\Gamma = \text{ИСТИНА}$  // проверка монотонности

для  $i$  от 1 до  $n-1$

нз

если  $(m[i, 2] \leq m[i+1, 2])$  то

$\Gamma = \text{ЛЖЬ}$

иначе // валид, выход из цикла

всё

кз

если  $(\Gamma = \text{ИСТИНА})$  то

вывод "функция является монотонно убывающей"

иначе

~~вывод~~ вывод "функция не является монотонно убывающей"

всё

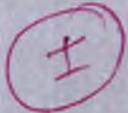
кон

→



ВНИМАНИЕ! Прочитайте только то, что записано с этой стороны листа в рамке справа

№ 3 *Матрица*



для алгоритма

- 1) *блески матрицы* (в матрице 0 - пусто, 1 - стена, 2 - ~~блеск~~ место для робота, 3 - цель)
- 2) *проходы массив* от  $[1, 1]$  до  $[m, n]$
- 3) *если вырывается* 2 или 3, *составить массив* *возв. координаты* *для робота.*
- 4) *если ячейка*  $[M, 1] = 2$ , *то обнаружить* *до чьей стены.*  
*анг. Олимпиада робота*  
*кат*

*чел M*

*чел N*

*блеск M*

*блеск N*

*чел атака [M, N] || 0 - пусто, 1 - стена,*

*чел графа [M, N] || 2 - место для робота*

*чел i от 1 до M || 3 - цель*

*иц || заполнить матрицу*

*чел j от 1 до N*

*иц, блеск атака[i, j]*

*кч*

*граф [1, N] = 3*

*(Продолжить решение стр 10)*



ВНИМАНИЕ! Прорезается только то, что записано с этой стороны листа в рамке справа

```

// лист 10
// (проход минус 1/3)
for (int i = 1; i <= M; i++)
    for (int j = 1; j <= N; j++)
        if (graph[i][j] == 2)
            if (graph[i][j] == 3)
                if ((i + 3 <= M) && (j - 1 >= 1) &&
                    (array[i + 3][j - 1] == 0))
                    graph[i + 3][j - 1] = 2;
            else
                if ((i + 1 <= M) && (j - 3 >= 1) &&
                    (array[i + 1][j - 3] == 0))
                    graph[i + 1][j - 3] = 2;
            else
                if ((i + 3 <= M) && (j + 1 <= N) &&
                    (array[i + 3][j + 1] == 0))
                    graph[i + 3][j + 1] = 2;
        else
            // (программное лист 11)

```



ВНИМАНИЕ! Прочисляется только то, что записано с этой стороны листа в рамке справа

(Задача 3) . лист 11

если  $(i+1 \leq M)$  и  $(j+3 \leq N)$  то

если  $(\text{array}[i][j+3] - [i+1, j+3]) \neq 0$  то

ли

ли

кч

кч

если  $(\text{graph}[M, 12] = 2)$  то

вернуть "yes"

иначе

вернуть "no"

ли

кон



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

√2

это лист 112

алг. уродн  
мат.

$$\text{лег. } z = 1 + (1 / (1 + (1 + (1/4))))$$

$$\text{лег. } r_1 = 2 + 1/z$$

$$z = 1 + (1 / (1 + (1 / (1 + (1 / (4 + z)))))))$$

$$\text{лег. } r_2 = 2 + 1/z$$

$$\text{ПОКА } \text{ABS}(r_2 - r_1) > 0,0001$$

К4

$$r_1 = r_2$$

$$z = 1 + (1 / (1 + (1 / (1 + (1 / (4 + z)))))))$$

$$r_2 = 2 + 1/z$$

К4

~~if~~ если  $(r_2 > 2,64575)$  и  $(r_2 < 2,64524)$  по  
вектор 'q'  
имеет вектор "no"

К.О.И

алгоритм:

Декартова лестница в зависимости  
структуры 1,1,1,4 прилеж  $r$  упрощенно

челл. Три этажа (структура) с самым малым  $\rightarrow$



ВНИМАНИЕ! Призерятся только те, что записано с этой стороны листа в рамке справа

Прог №2

увеличивается. Хозяину гостиницы  
считать только эти варианты до  
нес пер тека  $\Delta \leq 0,0001$

$$\sqrt{7} = 2.6452 \Rightarrow \text{Самый лучший}$$

вариант имеет в пределах от

$$2.64575 \text{ до } 2.64584 \text{ по формуле}$$

берем, иначе нет.

## Олимпиада школьников «Надежда энергетики»

И11F03	дистанционно с использованием ИКС
--------	--------------------------------------

№ группы

Место проведения

GR14-26
---------

шифр

— Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73111

ФАМИЛИЯ ВОЛКОВА

ИМЯ ИЯ

ОТЧЕСТВО ВЛАДИМИРОВНА

Дата рождения 22.04.2004

Класс: 11

Предмет Информатика

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на \_\_\_\_\_ листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: \_\_\_\_\_

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверьте только то, что записано с этой стороны листа в рамке справа

N3

Таблица истинности:

a	b	a → b
0	0	1
0	1	1
1	0	0
1	1	1

Записаны условия:

$$(A \rightarrow C) = 1 \quad C = 0$$

$$(A \rightarrow D) = 1 \quad D = 0$$

$$\downarrow$$

$$A = 0$$

$$(A \& B \rightarrow D) = 1$$

$$D = 0$$

$$\downarrow$$

$$A \& B = 0$$

(подходим, т.к.  $A = 0$ )

$$(\bar{B} \rightarrow E) = 1$$

$$\bar{B} = 0$$

$$B = 1$$

Ответ:  $A = 0$

$$B = 1$$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N 2

Будем вычислять нашу дробь

$$q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{1}{\dots}}}$$

Эта дробь будет близка к  $\frac{p}{q}$   
Вычисляем последовательно  $\frac{p_n}{q_n}$  пока не достигнем

до  $\sqrt{13}$ :

$$p_n = q_n p_{n-1} + p_{n-2}$$

$$q_n = q_n q_{n-1} + q_{n-2}$$

Будем последовательно вычислять  $\frac{p_n}{q_n}$ ,

пока не произойдет  $\left| \frac{p_n}{q_n} - \frac{p_{n-1}}{q_{n-1}} \right| < 0,00001$

(каждый раз погрешность уменьшается, как минимум  $0,0001$ )

Если в этот момент  $\left| \sqrt{13} - \frac{p_n}{q_n} \right| > 0,0001$ ,

то формула верна, иначе нет.

нач

цел  $c, p_0, q_0, p_1, q_1, p_2, q_2, q \in \mathbb{Z}, c \in \mathbb{S}$

$$q = [2, 1, \overline{2}, 1, 2, 8]$$

$$p_0 = q[0]$$

$$q_0 = 1$$

$$p_1 = q[0] \cdot q[1] + 1$$

$$q_1 = q[1]$$

$$p_2 = q[2] \cdot p_1 + p_0$$

$$q_2 = q[2] \cdot q_1 + q_0$$

$$c = 3$$

...



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N2 продолжение

пока  $\left| \frac{P_2}{Q_2} - \frac{P_0}{Q_0} \right| > 0,00001$

и y

$$Q_0 = Q_1$$

$$P_0 = P_1$$

$$Q_1 = Q_2$$

$$P_1 = P_2$$

$$Q_2 = q \text{ с } 3 \text{ } Q_1 + Q_0$$

$$P_2 = q \text{ с } 3 \text{ } P_1 + P_0$$

$$c = (c+1) \% 6$$

к y

если  $\left| \frac{P_2}{Q_2} - \sqrt{5} \right| > 0,0003$

вывод "формула неверна"

иначе

вывод "формула верна"

N5

Оптимизируем двумерный массив по формуле элемента y двух. Помогает перебор по всем вторым элементам.

Если найдем число, меньшее предыдущего, то функция не возрастает.

нач

цел x[n], y[n], count, n, i

ввод n

для i от 1 до n

и y  
ввод x[i]

к y

для i от 1 до n

и y  
ввод y[i]

к y



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N5 продолжение

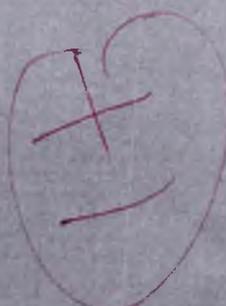
```

Quick-Sort-2 (x, y, arr, 1, n)
count = 0
for i from 2 to n
  ny
  если  $y[i] \leq y[i-1]$ 
    count = count + 1

  ny
  если count > 0
    вывод " Функция не Функция не монотонно
    возрастающая"
  иначе
    вывод " Функция монотонно возрастающая"
кон

arr QuickSort2 (x, y, n1, n2)
наз
  var i, j, k, t, s
  k =  $(n1 + n2) \text{ div } 2$ 
  i = n1
  j = n2
  повтори
    пока  $x[i] < k$ 
      ny
      i = i + 1
    ny
    пока  $x[j] > k$ 
      ny
      j = j - 1
    к'ы
    если j < i то
      t =  $x[i]$ 
       $x[i] = x[j]$ 
       $x[j] = t$ 
      ...

```





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N5 продолжение

...  
 $s = y \text{ с } i \text{ з}$   
 $y \text{ с } i \text{ з} = y \text{ с } j \text{ з}$   
 $y \text{ с } j \text{ з} = s$   
 $i = i + 1$   
 $j = j - 1$

иначе ~~continue~~

если  $i = j$  то

$i = i + 1$

$j = j - 1$

в с i

в с i

go ← относится к "невыпрямле"

если  $n_1 < j$  то

QuickSort2(x, y, n1, j)

если  $i < n_2$  то

QuickSort2(x, y, i, n2)

кон

N4

Будем идти по левой нижней клетке в соответствии со стрелками в радиусе:  
 1 (вверх), 2 (вправо), 3 (вниз), 4 (влево) - эти числа будут записаны в массиве x.

Останавливаемся, когда выйдем в (N, N)  
 На каждой пройденной клетке будем вычислять значение и соответств. число по массиву y простого. Для этого предварительно создадим массив из чисел от 1 до 1000000 (можно больше, но должно хватить) с помощью решета Эратосфена. Если число простое, то оно останется собой, если составное или 1, то на



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

14 продолжение  
Во месте стоим 0  
Пройди по всем клеткам по стрелкам, выводи  
только простые числа.

ans Resh()

цел n = 1000000, x[13]

где i от 1 до n

кы

x[i] = i

кы

x[13] = 0

пока i <= n

кы

если x[i] != 0

j = i + 1

пока j <= n

кы

x[j] = 0

j = j + 1

кы

i = i + 1

кы

вывод x

~~вывод~~

кон

Здесь написано решение Эратосфена. Это работает у  
1000000 чисел. Если на i-ом месте 0, значит i  
составное, иначе i простое.

x - массив

y - сами числа

на z

цел N, i, j, x[i], j, y[i], j,

где i от 1 до N

где j от 1 до N

вывод x[i], j



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N4 продолжение

```

a = Aesh()
i = 1 j = 1
пока i < N or j < N
нч
  если a [y[i], j] = 0
    ввож y[i], j
    если x[i, j] = 1
      i = i + 1
    если x[i, j] = 2
      j = j + 1
    если x[i, j] = 3
      i = i - 1
    если x[i, j] = 4
      j = j - 1

```



кч

квн

N3

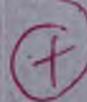
Будем решать задачу с помощью рекурсии. Введем ф-цию, которая вычисляет, возможно ли прийти в данную клетку, используя грани и возможности появления в клетках, у которых мы можем попасть в данную.

```

алз
нат
где M, N, A ∈ M, N, j, i
ввож M
ввож N
где i от 1 до N
нч
где j от 1 до M
нч
ввож A ∈ M, N
кч
кч
если recurs(N, M)
  ввож "Da"
...

```

здесь 1, если элемент  
0, если ei нет





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3 продолжение

```

...
иначе
  bobog "Нет"
кон
алг Recurr (x, y)
  итер counter
  если x == 1 and y == 1
    bobog True
  иначе
    если x > N or y > M or x < 1 or y < 1
      bobog False
  иначе
    если если Recurr(x-1, y-1) + Recurr(x+1, y-1)
    иначе + Recurr(x-1, y+1) + Recurr(x+1, y+1)
    иначе bobog True
  иначе
    если A(x, y) == 1
      a = x-1, b = y-1
      итер i от 1 до N
      итер если A(a, i) == 1 and
        Recurr(a, i) + Recurr(a, i+1) +
        + Recurr(a, i+2) + Recurr(a+1,
        i+2)
        bobog True
      итер
    если A(x, y) == 1
      a = x+1, b = y-1
      итер i от 1 до N
      итер если A(a, i) == 1 and Recurr
        (a, i) +
        + Recurr(a, i+1) + Recurr(a, i+2) +
        + Recurr(a-1, i+2)
        bobog True
      итер
  иначе
    если A(x, y) == 1
    ...

```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3 задание

...  
 $a = x - 3, b = y - 3$   
где  $i$  от  $0$  до  $N$

мы  
если  $A[x, y] = 1$  and  $\text{Recurs}(a, i) +$   
 $\text{Recurs}(a+1, i) + \text{Recurs}(a+1, i+1)$   
 $+ \text{Recurs}(a+1, i+2)$

return True

мы

~~return~~

~~return A[x, y]~~

иначе

если  $A[x, y] = 1$   
 $a = x + 3, b = y - 3$   
где  $i$  от  $1$  до  $N$

мы  
если  $A[x, y] = 1$  and  $\text{Recurs}(a, i) +$   
 $\text{Recurs}(a+1, i) + \text{Recurs}(a+2, i) +$   
 $\text{Recurs}(a+2, i-1)$

return True

мы

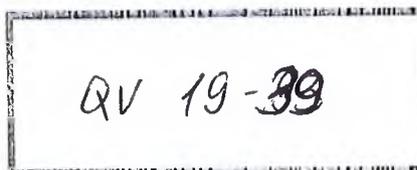
кон

# Олимпиада школьников «Надежда энергетики»

II OF 02	Дистанционно, с использованием ВКС
----------	---------------------------------------

№ группы

Место проведения



шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73101

ФАМИЛИЯ Горбаченко

ИМЯ Кирилл

ОТЧЕСТВО Александрович

Дата рождения 04.12.2004

Класс: 10

Предмет Информатика

Этап: Заключительный

Работа выполнена на 7 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Задача 1

$$A \rightarrow C = 1$$

$$C = 0$$

$$A - ?$$

$$A \& B \rightarrow D = 1$$

$$D = 0$$

$$B - ?$$

$$\bar{B} \rightarrow E = 1$$

$$E = 0$$



Заметим, что  $A \rightarrow C = 1$ , т.к.  $C = 0$ , то  $A \rightarrow 0 = 1$ , но  $A \rightarrow 0$  - истина, если  $A = 0 \Rightarrow A = 0$

Аналогично,  $\bar{B} \rightarrow E = 1$ ,  ~~$\bar{B} \rightarrow E = 0 \Rightarrow \bar{B} \rightarrow 0 = 1$~~

$$\Rightarrow \bar{B} = 0 \Rightarrow B = 1$$

Заметим, что  ~~$\bar{A}$~~  при  $A = 0$  и  $B = 1$   $A \& B \rightarrow D = 1$ , также выполняется. Действительно,  $A \& B = 0 \& 1 = 0$ ,  $D = 0$   
 $\Rightarrow A \& B \rightarrow D = 0 \rightarrow 0 = 1$

$$\text{Ответ: } A = 0; B = 1$$

*истина?*



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

### Задача 5

где?

Отсортируем таблицу по возрастанию координаты  $x$ .

~~Пройдем по ней сверху, в и на каждом ходу будем~~

Создадим переменную  $p$  и дадим ей искомое значение =  $x$  координаты с минимальным  $x$  (т.е. <sup>первым столбике</sup>

$y$  в ~~первой строке~~ таблицы)

Начнем обход массива ~~уже~~ со второго элемента и на каждом ходу будем проверять, меньше ли текущий  $y$ , чем ~~предыдущий~~ ~~элемент~~ ~~этого столбца~~, ~~тогда обновим~~ ~~предыдущий~~ элемент предыдущий  $y$  (т.е.  $p > y_{текущий}$ ) если  $y$  меньше

обновим  $p$ , ~~и~~ <sup>те</sup> обновим ему значение текущего  $y$  ( $p = y_{текущий}$ )

~~если же~~ и продолжим обход ~~таблицы~~, пока

~~заканчива~~ <sup>заканчиваем</sup> выводим, что функция не монотонно убывающая, и ~~продолжим~~ ~~выполнение~~ программы.

После выполнения обхода, ~~т.к.~~ программа не закончила свою работу <sup>раньше</sup> это значит, что функция монотонно ~~убывает~~ убывающая. Выведем, что функция монотонно убывающая и закончим выполнение программы.

Т



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

### Задача 3

Пусть  $tr$  - карта зала,  $s_x$  и  $s_y$  - соответствующие координаты начальной позиции робота, а  $f_x$  и  $f_y$  - соответствующие координаты конечного положения робота (т.е. места в котором он должен прийти)

Тогда заполним карту  $tr$  так: если в клетке зала с координатами  $(i, j)$  стоит колонка, то  $tr[i][j] = true$ , иначе  $tr[i][j] = false$ .

Решим алгоритм проверки, может ли робот дойти до конечной позиции.

```

bool dfs(int curx, int cury) { // соответственно текущие
    // координаты положения
    // робота
    if (curx == f_x && cury == f_y) { // если текущая
        return true; // позиция = конечной
    } // возвращаем true.
    if (curx + 3 < n && cury + 1 < m) { // проверка, что ход
        // не выйдет за границы
        // карты
        bool br = false; // br - хранит true или false в зависимости от того
        // вернется ли робот по пути на ходу или не вернется
        // т.к. изначально колонки еще не вернулись, то br = false
        for (int i = 0; i < 4; ++i) { // циклом проходим
            if (tr[curx + i][cury + 1]) { // все клетки на по которым
                br = true; // идет ход робота и
                break; // проверяем есть ли в ней
            } // колонка
        }
        if (!br) { // т.е. колонки на пути нет
            return false;
        }
        if (dfs(curx + 3, cury + 1)) { // делаем ход и проверяем
            // еще из новых координат
            // есть ли из нее есть путь
            // и из нашей клетки есть
        }
    }
}

```



ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа

Аналогично для трех других траекторий ходов. В конце функции  $dfs$  напишем  $return false$ ; т.е. если ~~ни одна~~ из 4 траекторий не ведет в конкретную клетку, то и наша клетка не ведет.

Тогда ~~мы~~ заменив ~~то~~ ~~зачислим~~ функцию  $dfs$  передав ей локальные координаты, т.е.

```

if (dfs(stx, sty)) {
    cout << "yes";
} else {
    cout << "no";
}

```

т.е. если  $dfs$  вернет и true, то путь из начальной и в конечную есть, поэтому ~~и~~ программа выведет yes и если же путь не найден, то она ~~и~~ выведет no

Заметим, что программа не будет выполняться бесконечно. Действительно, программа будет выполняться бесконечно, если есть путь из одной клетки в себя же за какое-то количество ходов, но такого не может быть, т.к. любая траектория ходов ~~то~~ увеличивает координату  $y \Rightarrow$  она постоянно растет, но для возвращения в ту же клетку ~~она~~ координата  $y$  должна стать прежней, т.е. уменьшиться, что не может быть  $\Rightarrow$  программы не будет выполняться бесконечно и когда-нибудь выдано ответ.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

### Задача 4

Функция ~~вернет~~ ~~boolean~~ ~~значение~~ ~~boolean~~, которая  
 Пусть  $sinx$  и  $siny$  - ~~на~~ ~~текущее~~ ~~положение~~ Рюка,  
 изначально равно соответственно ~~каким-то~~ ~~нечисловым~~ ~~координатам~~  
 положения Рюка. Также  $finx$  и  $finy$  - соответственно  
 координаты конечной клетки лабиринта.  $map$  - карта  
 самого лабиринта со стрелками, по которым необходимо  
 двигаться. ~~apple~~ - ~~текущее~~ ~~яблоко~~, ~~по~~ ~~которому~~ ~~Рюк~~  
~~тогда~~ ~~apple~~ ~~является~~ ~~яблоком~~, ~~которое~~ ~~apple~~ - ~~команда~~  
 написанное на текущем яблоке, повешенная Рюку.  
 Тогда ~~apple~~ массив всех ~~номеров~~ ~~яблоков~~, которые взял Рюк  
 Тогда ~~apple~~.

~~while (sinx != finx || siny != finy) {~~  
 // цикл закончился  
 // только тогда,  
 // когда Рюк достиг  
 // последней  
 // клетки

~~Будет ходить по лабиринту пока sinx != finx || siny != finy~~

Пусть  $pr$  - массивы из  $bool$  и  $bool$ , где если  
 $pr[i] = true$ , то  $i$  - простое число.  
 Инициализируем массив  $pr$  в диапазоне решето  
 Эратосфена, до какого-то большого числа (например  $N \times N$ ,  
 где  $N$  - сторона лабиринта), изначально все значения  $= true$ ;

~~Тогда решето Эратосфена, изначально все значения~~  
~~pr[i] = true;~~  
~~pr[0] = false;~~  
~~pr[1] = false;~~

Тогда решето Эратосфена:  
~~void res~~ void ~~resh(int n)~~ //  $n$  - ограничение, до куда  
 // целить решето  
~~re~~  $pr[0] = false;$  // 0 - не простое  
 $pr[1] = false;$  // 1 - не простое  
~~for (int i = 2; i <= n; ++i) {~~  
~~for (int j =~~



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

if (pr[i] != 0) {
    // если i еще простое
    for (int j = i + i; j < n; j += i) {
        pr[j] = false; // т.к. j попадает при проходе
        // всех чисел (до n) кратных i, то
        // j - кратно i, а т.к. i != 1 и j != i, =>
        // j - не простое
    }
}

```

Заполнив pr начнем обход лабиринта

```

while (curx != finx || cury != finy) {
    // пока обе координаты
    // положения Рюка не
    // равны финальной точке,
    // т.е. Рюк еще не вышел,
    // цикл выполняется

    if (pr[apple])
        // т.е. в массиве pr
        // хранится значение 1
        // и apple (текущий номер
        // клетки) простым числом
        // не является
        continue;

    if (pr[apple] == 0) {
        apples.push_back(apple);
        // если apple - простое
        // вставляем его в
        // массив apples
        // т.е. Рюк от берега
    }
}

```

if (apple < n)
if (pr[apple]) {
if (pr[apple]) {
apples.push\_back(apple);
}
} else {
// если же apple >= n, то мы не можем двигаться
// и оно простым, тогда выполним решение go
// т.е. go apple и проверим, является ли оно
// простым

```

resh(apple);
if (apple != pr[apple]) {
    apples.push_back(apple);
}
}

```



// после выполнения необходимого движения по стрелке,
// находящейся в клетке, где мы находимся сейчас
// находится Рюк, т.е. изменяем curx или cury ?

// таким образом Рюк получит все яблоки с простым числом



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

## Задача 2

Посчитаем значение числа равно  $\sqrt{7}$  и запишем.

Далее начнем считать правую часть.

Пусть  $s_{nr}$  и  $p$  - текущее и предыдущее значения для правой части

~~Изначально~~ ~~используем~~ изначально  $p=0$

$$s_{nr} = 2$$

(7)

Будем вычислять алгоритм, пока разница между текущим и предыдущим  $(s_{nr} - p) > 0,0001$ .

На каждом ходу будем считать правую часть к глубины на 1 больше предыдущей. Изначально глубина 50.

Пусть глубина =  $n$ , сколько раз мы углубимся в знаменателе дроби ~~при~~ при чем погрузившись до ~~определенной~~ <sup>нужной</sup> ~~нужной~~ <sup>нужной</sup> глубины ~~разницы~~ <sup>или 4 раз</sup> как  $\frac{1}{4}$ , а дробь, с которой  $\frac{1}{4}$  складывается, заменим на  $1$  или  $\frac{1}{4}$  в зависимости от того, чему равно ~~число~~ ~~число~~ в следующей знаменателе, которое на первом месте.

Выполняем вычисление до нужной глубины для текущего числа и проверяем, что разница между текущим и предыдущим больше  $0,0001$ . Если больше, то продолжаем вычисление, ~~иначе~~ прибавив перед этим  $p$  новое значение, равное  $s_{nr}$ . Если же меньше, то проверяем если разница между  $\sqrt{7}$  и текущим числом, которое мы посчитали, по модулю больше  $0,0001$  выводим, что формула неверна, иначе ~~выводим~~ если разница меньше или равна  $0,0001$  выводим, что формула верна.

# Олимпиада школьников «Надежда энергетики»

I11F02	Автоматизация с использованием ВКС
--------	------------------------------------

№ группы Место проведения

GR 98-84
----------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73111

ФАМИЛИЯ ЕГОРОВ

ИМЯ МАКСИМ

ОТЧЕСТВО ЮРЬЕВИЧ

Дата рождения 01.07.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: Заключительный

Работа выполнена на 8 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

~ 1

$$A \rightarrow C \equiv 1$$

$$\bar{A} \vee C \equiv 1, \bar{A} \vee 0 \equiv 1 \Rightarrow A \equiv 0$$

$$\bar{B} \rightarrow E \equiv 1, \bar{B} \vee E \equiv 1, B \vee E \equiv 1, B \vee 0 \equiv 1 = B \equiv 1 (\neq)$$

$$A \wedge B \rightarrow D \equiv 1, (\bar{A} \wedge \bar{B}) \vee D \equiv 1$$

$$\bar{A} \vee \bar{B} \vee D \equiv 1, \bar{A} \vee \bar{B} \vee 0 \equiv 1, \text{ (не выполняем)}$$

$$\bar{A} \vee \bar{B} \equiv 1, \text{ т.к. } A \equiv 0, B - \text{любое (х-любое)}$$

Ответ: A - истина; B - истина

~ 4 Алг шимки

max 10г  $\text{возможное} = \text{true}$

цел  $x, y, i, j, N, \text{ans} = 0$ , направление = 1

цел  $\text{use} [N, N]$ ,  $\text{use} [2]$

$\text{use} [1] = 1$ ,  $\text{use} [2] = -1$

for  $N$

если  $N \leq 0$  то

возврат ("некорректные входные данные")

иначе

для  $i$  от 1 до  $N$

для  $j$  от 1 до  $N$

if  $\text{use} [i, j]$

ку

ку

$x = N, y = 1$

если  $\text{use} [x, y]$  то

$\text{ans} = \text{ans} + 1$

все

пока  $x > 0$

ку

(примечание: не ссы-  
латься)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

для  $i$  от 0 до 1

$x = x - 1$

если  $x > 0$  и простое (или  $\{x, y\}$ )

$ans = ans + 1$

все

мы

если  $x \leq 0$  и  $y < N$  то

возвращаем False

все

если не  $(x = 1$  и  $y = N)$  то

для  $i$  от 0 до  $N - 2$

мы  $y = y + \text{хотим ли направление?}$

если  $x > 0$  и простое (или  $\{x, y\}$ ) то

$ans = \text{ans} + 1$

все

мы

все

направление = 3-направление

мы

если возможно

возврат ("горизонтальная вершина правой ветки листа")

возврат ("линия сверху - ",  $ans$ )

иначе

возврат ("горизонтальная вершина левой ветки листа")

все

или

ЛОТ АЛГ простое (или  $y$  или  $N$ )

иначе

если  $N < 2$  или  $(N > 3$  и  $N \bmod 2 = 0)$  то

возврат False

все

для  $i$  от 3 до  $\sqrt{N}$  шаг 2

мы

если  $N \bmod i = 0$  то

возврат False

все

мы

или





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

### N5 АЛГ МОНОТОННОСТЬ

нач

yes  $N, i$   
no монотон = True, возможно = True

for  $N$

если  $N \leq 0$  то

return ("некорректные входные данные")

иначе

for  $x \in N, y \in N$

для  $i$  от 1 до  $N$

do for  $x[i]$

for  $y[i]$

do Quick Sort ( $x, y, 1, N$ )

$i = 2$

while  $i \leq N$  и монотон и возможно

do если ~~True~~  $y[i] \leq y[i-1]$  то  
монотон = false

for  $x[i] = x[i-1]$  то  
возможно = false

for  $i = i + 1$

if не возможно то

return ("функция не монотонна")

return ("для одного  $x$  больше одного  $y$ ")

иначе если монотон то

return ("функция монотонно  
возрастает")

иначе

return ("функция не монотонно  
возрастает")

for  
for

кон

(использована на след. листе)





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

АЛГ Quick SORT (~~array, left, right~~)  
 arr beg x [ ], arr beg y [ ], arr yem Left, arr  
 yem Rights)

нел  
 yem L, R

L = left, R = rights

beg pivot = x [ (L+R) div 2 ]

~~если~~ если L < R то

ПОКА L ≤ R

нел

ПОКА x [ L ] < pivot

нел

L = L + 1

нел

ПОКА x [ R ] > pivot

нел

R = R - 1

нел

если L ≤ R то

swap (x [ L ], x [ R ])

swap (y [ L ], y [ R ])

L = L + 1, R = R - 1

конец

нел

Quick SORT (x, y, left, R)

Quick SORT (x, y, L, rights)

конец

нел

АЛГ swap (arr beg a, arr beg b)

нел

beg temp

temp = a

a = b

b = temp

конец



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

~~Анализ~~ из АЛГ ПОПРЫГУМ

```

bool
ум M, N, i, j
bool симметрична [M, N], всецело [M, N]
bool M, N
или M <= 0 или N <= 0 то
bool ("некорректные базисные данные")
иначе
  для i от 1 до M
  для j от 1 до N
  bool симметрична [i, j] # TRUE - симметрична
  всецело [i, j] = False // False - ПЛИТА
  или
  симметрична [1, N] = TRUE
  симметрична [M, 1] = TRUE
  if S (симметрична, всецело, M, 1, M, N)
  или всецело (1, N) то
    bool ("верная правая шина
    соединена")
  иначе
    bool ("верная правая шина
    не соединена")
  все:
  все:
  кон
  (продолжение на след. листе)

```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Алг dfs (arr loc g, J, arr loc e used, J,  
arr loc x, arr loc y, arr loc m, arr loc n)

неч

arr loc x<sub>1</sub>, x<sub>2</sub>, y<sub>1</sub>, y<sub>2</sub>

arr loc x<sub>0</sub>[1], x<sub>0</sub>[2]

x<sub>0</sub>[1] = 1, x<sub>0</sub>[2] = -1

неч временноВерн = False, временноТор = False

x<sub>1</sub> = x, y<sub>1</sub> = y

~~неч временноВерн~~

если 1 ≤ x и x ≤ m и 1 ≤ y и y ≤ n и не

used[x, y] то

used[x, y] = True

и не временноВерн

неч

если g[x<sub>1</sub>, y<sub>1</sub>] то

временноВерн = True

неч

счит dir от 1 до 2

неч

y<sub>2</sub> = y<sub>1</sub> + x<sub>0</sub>[dir]

x<sub>2</sub> = x<sub>1</sub>

временноТор = False

если 1 ≤ y<sub>2</sub> и y<sub>2</sub> ≤ n и не временноТор

неч

если g[x<sub>2</sub>, y<sub>2</sub>] то

dfs(x<sub>2</sub>, used, x<sub>2</sub>, y<sub>2</sub>, m, n)

временноТор = True

все

y<sub>2</sub> = ~~y<sub>2</sub>~~ y<sub>2</sub> + x<sub>0</sub>[dir]

неч

если не временноТор то

y<sub>2</sub> = y<sub>2</sub> - x<sub>0</sub>[dir]

(продолжить на след. листе)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

если  $x_2 \leq M$  и не ~~выпущено~~  $\text{Top}$

если  $g[x_2, y_2] \neq 0$

$\text{dfs}(g, used, x_2, y_2, M, N)$

$\text{выпущено} \text{Top} = \text{True}$

все

$x_2 = x_2 + 1$

и

все

и

все

$x_1 = x_1 - 1$

и

все

кон

~ 2 АЛГ ~~КРАТКАЯ ЗАДАЧА~~

АЛГ Выражение

нел

$\text{eps} = 0.0001$

$\text{res} = \text{sqrt}(19)$

~~seq = 1~~

и  $\text{seq}[0]$

$\text{seq}[1] = 2, \text{seq}[2] = 1, \text{seq}[3] = 3$

$\text{seq}[4] = 1, \text{seq}[5] = 2, \text{seq}[6] = 8$

если  $\text{abs}(\text{res} - \text{calcSum}(\text{seq}, 0, 6, 1000)) \leq \text{eps}$

то

$\text{bool}(\text{"выражение верно"})$

иначе

$\text{bool}(\text{"выражение не верно"})$

кон

все

(продолжение на след. листе)



ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа



Возьмем  $arr$  calcSum(arr) и  $seq[i]$ , и  $max$ ,  
и  $size$ , и  $precision$

нач

и  $seq$  ~~ans~~  $ANS = seq[i]$  и  $max$  и  $size + 1$   
если  $max < precision$  то

~~ans~~  $ANS = ANS + 1.0$  / calcSum(seq,  
 $max + 1, size, precision$ )

и

возвращаем ANS

кон



# Олимпиада школьников «Надежда энергетики»

III F03	ДИСТАНЦИОННО, С ИСПОЛЬЗОВАНИЕМ ВКС
---------	---------------------------------------

№ группы

Место проведения

GR 14-68
----------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 7311

ФАМИЛИЯ ЗЕНИН

ИМЯ МИРОН

ОТЧЕСТВО АЛЕКСАНДРОВИЧ

Дата рождения 21.06.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на \_\_\_\_\_ листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: Зенин

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

~1  
 $C, D, E = 0$

$\bar{B} \rightarrow E = 1$



$B \vee E = 1$



$B \vee 0 = 1$

что возможно только при  $B = 1$ .

~~4~~

~~Предположим, что...~~

~5

Для этого следует переформулировать таблицу по X-y. Заметим, что...  
Заменим переменные, выразим...  
используя boolean переменные и числа  $(y_n - y_{n-1})$ .  
смотрим, имеют ли все промежуточные выражения вид...  
это и у числа (или 0)?

~6

Для этого следует переформулировать таблицу по X. Заметим, что...  
используя две boolean переменные (одна - для определения знака, другая - для знака  $(y_n - y_{n-1})$ ).  
Если знак определен, по узловой паре  $(y_n - y_{n-1})$  знак не совпадает с определенным знаком (а именно +, ведь иначе это не будет монотонно возрастающей функцией, чтобы описать этот момент), но ф-я не является монотонно возр (или  $y_n - y_{n-1}$  и определенная, как и при определении знака).

Тогда алгоритм следующий:

Ответ:  $A = \text{ложь} (0)$   
 $B = \text{ложь} (1)$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

~ 5

Для того чтобы эффективно использовать таблицу по массиву  $x$  в возр. порядке. Далее пройдем по таблице, покажем разности пар  $(x_n - x_{n-1})$ . Если она отрицательна -  $\varphi$ -я не монотонно возрастает. Алгоритм формирования массива  $(x_{n+1} = 0 \text{ (или } \log n))$  в массиве  $(x_{n+1})$  (индексация в массиве с 0!)

сформировать (массив,  $om, go$ )

если  $om \geq go$  рассмотрим слева элементы, которые меньше справа (первый элемент, индекс 0), а справа те, которые равны или больше, и записать в переменную  $p$  индекс первого положительного элемента.

сформировать (массив,  $om, p-1$ )

сформировать (массив,  $p+1, go$ )

Сортировку проведем по массиву  $[i].x$ !!!

Собственно, программа:

массив = получить массив разниц (элементы, что если  $N=1$ , сразу вывести сортировку и вывести массив проверки. Если  $N \leq 0$ , куда-то амидуг)

сформировать (массив, 0, N)

for  $i = 0$   $i < N$   $i++$

если массив  $[i].y$  - массив  $[i-1].y < 0$  вывести сообщение  $\varphi$ , "проблема" (т.е.  $\varphi$ -я не монотонно возрастает) вывести из программы

вывести сообщение об успехе (т.е.  $\varphi$ -я монотонно возрастает)

~ 2

$$\sqrt{19} = 4 + \frac{1}{2 + \dots}$$

сохраняется эта запись

$$\left| \sqrt{19} - 4 - \frac{1}{2 + \dots} \right| = \text{погрешность}$$

Заметим, что при добавлении слагаемых погрешность уменьшается, и само выражение  $\frac{1}{2 + \dots}$  монотонно. В новом выражении погрешность при  $\sqrt{19}$ -ом разе слагаемых будет уменьшаться (погрешность при  $\sqrt{19}$ -ом разе слагаемых  $|\sqrt{19} - 4 - \frac{1}{2}| = \frac{1}{3}$ , а при разе  $|\sqrt{19} - 4 - \frac{1}{2}| = \frac{1}{3}$ , что меньше того, что было при слагаемых).



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

В таком случае оптимальный алгоритм такой:  
~~Решается как в~~  
Увелич. как-то ширину до тех пор, пока по-  
решимость не станет ~~и~~ меньше-равно 0,0001. ~~Вывод~~

```

Вывести ширину до тех пор, пока по-решимость не станет <= 0,0001. Храним
всё, что нужно, в массиве double
Собственно программа:
структура = {2, {1, 5, 1}, 2, 8}
for i = 1 to i++ (н.е. цикл без ограничения. Выводим из него
пока не достигнем условия)
    ширина = 0
    for i2 = i to i2 > 0 i2 --
        ширина = 1 / (структура[(i2-i) % структура.разм] + ширина)
    если |sqrt(3) - 4 - ширина| <= 0,0001
        вывести i (н.е. как-то неаккуратных структур)
        вывести из программы

```

№4 ~~Решается как в~~ (массив содержит)

Пусть массив содержит следующие значения:  
 вверх значение индекса  
 вниз и  
 влево (направление и др.)  
 вправо

Они соединены в одну направленную цепочку.  
 Сначала создадим алгоритм передвижения (окажем, что  
 походу по симметричной правой ветке элемент).

массив = массив массивов (если N=1, вывести, если  
 иначе, индекс (с 0 до N-1) завершить работу) ! = < не равно

```

x = 0, y = N-1
пока x != 0 и y != 0
    если массив[x][y] == 'вверх' (массив[x][y] == 'вверх')
        вывести массив[x][y].инд
        вывести вариант из массива[x][y].инд
        вверх: y --
        вниз: y ++
        влево: x --
        вправо: x ++

```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

если простое  $m$  (максимум  $\lfloor \frac{N}{m} \rfloor$  раз)  
вывести максимум  $\lfloor \frac{N-1}{m} \rfloor$  раз

~~Заметим, что если  $n$  делится на  $m$ , то  $n/m$  делится на  $m$~~

Теперь сделаем вывод о том, определяющую роль играет "число". Алгоритм "пробит" до безобразия:

простое  $m$  ( $n$ )

~~если  $n \% m = 0$  то~~

если  $n = 1$  (1 - не простое число!!!)

вернуть 1

если  $n < 20$

for  $i = 2$   $i < n$   $i++$

если  $n \% i = 0$

вернуть 1

~~for~~ (то, на какое количество  $n$  делится, вычислим  $n$  раз)  
если  $n \% [2, 3, 5, 7, 11, 13, 17, 19] = 0$

вернуть 1

(теперь мы знаем, что если  $n$  не простое, то  $n = k \cdot m$ , где  $k$  и  $m \geq 20$  (примем, при любой группировке). Поэтому вернем промежуточные результаты:

for  $i = 20$   $i \leq n/20$  (окружно)  $i++$

(ведь вычислить не можем, ведь  $k$  и  $m \geq 20$ )

если  $n \% i = 0$

вернуть 1

вернуть 1

используя тот момент более подробно. Пусть  $k_{min} < \frac{n}{k_{min}}$ . Но ведь  $\frac{n}{20} > \frac{n}{k_{min}}$ , а  $\frac{n}{k_{min}}$  — максимальный

возможный делитель

~ 3

Будет идти "концы". Тогда все варианты обратного

вычисления:

1 влево 3 вправо

1 вправо 3 влево

3 влево 1 вправо

3 вправо 1 влево

Заметим, что при каждом ходе мы идем влево, значит, достаточно применить алгоритм рекурсии:

7



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Сначала заменим сигнатуру матрицы:

$M$  - ам-во шагов,  $N$  - широта.

Каждый шаг ~~состоит~~ является - лев. зр. или, наоборот, но сигнатура здесь есть, или нет - неважно.

Решение

матрица  $(\text{матрица}, x, y, M, N)$

если  $x=0$  и  $y=N-1$   
вернуть истину

for  $i=0$   $i < 4$   $i++$

выбрать вариант  $i$

0: если выйдешь  $(x-1, y+3, M, N)$

матрица  $\text{матрица}$  = матрица  $(\text{матрица}, x-1, y+3, M, N)$

1: если выйдешь  $(x+1, y+3, M, N)$

матрица  $\text{матрица}$  = матрица  $(\text{матрица}, x+1, y+3, M, N)$

2: если выйдешь  $(x-3, y+1, M, N)$

матрица  $\text{матрица}$  = матрица  $(\text{матрица}, x-3, y+1, M, N)$

3: если выйдешь  $(x+3, y+1, M, N)$

матрица  $\text{матрица}$  = матрица  $(\text{матрица}, x+3, y+1, M, N)$

если матрица  $\text{матрица}$

вернуть истину  $\text{матрица}$

Ошибки  $\varphi$ -то выйдешь  $(x, y, M, N)$ :

выйдешь  $(x, y, M, N)$

если  $x < 0$  или  $y < 0$   
вернуть ложь

если  $x \geq M$  или  $y \geq N$   
вернуть ложь

вернуть истину

сигнатура  
 $x$  и  $y$  = сигнатура ~~сигнатура~~  
 $(\text{матрица}, x, y, M, N)$   
если  $x \neq y$  = null  
вернуть ложь  
иначе  
 $x = x \cdot y \cdot x$   
 $y = x \cdot y \cdot y$



ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа

и функцию  $xy$  (таблица,  $x, y, M, N$ ):

функция (таблица,  $x, y, M, N$ )

$xy = null$

for  $i=y$   $i \geq 0$  --

если  $xy[x][i]$

$xy = new xy(x, i)$

вернуть  $xy$

$xy.y$   $xy.x$

вернуть  $xy$

Алгоритм:

таблица = получить массив значений

$N$  = ~~получить~~  $xy.y$

$M$  =  $xy.x$

если  $xy$  (таблица,  $M-1, 0, M, N$ )

вывести содержимое  $xy.y$

иначе

вывести содержимое  $xy.x$

## Олимпиада школьников «Надежда энергетики»

Инфо	Дистанционно, с использованием ВКС
№ группы	Место проведения

GR 71-40
----------

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

Вариант № \_\_\_\_\_

ФАМИЛИЯ ИВАНЧЕНКО

ИМЯ БРОНИСЛАВ

ОТЧЕСТВО ЕГОРОВИЧ

Дата рождения 22.11.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 8 листах

Дата выполнения работы: 28.02.2021  
(число месяц, год)

Подпись участника олимпиады: \_\_\_\_\_

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№1.

Подставим значения C, D, E из условия, получим (0 - ложь, 1 - истина):

$$1.) A \rightarrow 0 = 1$$

$$2.) A \wedge B \rightarrow 0 = 1$$

$$3.) \neg B \rightarrow 0 = 1$$

Из третьего утверждения понимаем, что  $\neg B \neq 1$ , т.к. в таком случае утверждение было бы ложным  $\Rightarrow \neg B = 0 \Rightarrow B = 1$ .

Из первого понимаем, что  $A \neq 1$ , т.к. в этом случае утверждение было бы ложным  $\Rightarrow A = 0$ .

Подставим  $A = 0$  и  $B = 1$  во второе утверждение:  $0 \wedge 1 \rightarrow 0 = 0 \rightarrow 0 = 1$  - верно  $\Rightarrow B = 1, A = 0$

Ответ:  $A = 0, B = 1$



используйте логичность!

№3-кест



ВНИМАНИЕ! Проверится только то, что записано с этой стороны листа в рамке справа

√2

алг rec (арг вещ ~~рез~~ x)

Нач

$$x = x + 8$$

$$x = 1/x + 2$$

$$x = 1/x + 1$$

$$x = 1/x + 3$$

$$x = 1/x + 1$$

$$x = 1/x + 2$$

$$x = 1/x$$

КОН

алг Раскопки

Нач

вещ prev, res, sqrt19, diff

$$\text{sqrt19} = \text{sqrt}(19)$$

$$\text{prev} = 0.0$$

$$\text{res} = \text{rec}(0.0)$$

ПОКА  $\text{res} - \text{prev} \leq -0.0001$  ИЛИ  $\text{res} - \text{prev} \geq 0.0001$ 

НЧ

$$\text{prev} = \text{res}$$

$$\text{res} = \text{rec}(\text{res})$$

КЧ

$$\text{res} = \text{res} + 4$$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№2 (продолжение)

diff = sqrt10 - res

если diff > -0.00001 и diff <= 0.00001 то  
вывод "Верно"

ИНАЧЕ

ВЫВОД "НЕВерно"

ВСЁ

КОН

№5.

// Будем считать, что в цикле "пока"  
переменная будет изменяться вкратце  
но от первого значения и до второго,  
и что в массиве индексация начинается с 0.  
алг МоноТОННАЯ ФУНКЦИЯ

НАЧ

ЦЕЛ n

~~ЦЕЛ a[10000]~~

ВВОД n

ЦЕЛ a[n][2]

Для i от 0 до n-1

НУ

ВВОД a[i][0], a[i][1]

КУ

sort(a, 0, n-1)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

✓5 (продолжение)

Для  $i$  от 1 до  $n-1$

НЧ

если  $a[i-1][1] \geq a[i][1]$  то

Вывод "НЕ монотонно возрастающая"

Выход(0) // выход из программы, как return

всё

~~можно использовать~~

КЦ

Вывод "Монотонно возрастающая"

КОН

алг sort (арг рез цел  $x[n][2]$ , арг цел  $l$ , арг цел  $r$ ) // Quick Sort

НЧ

цел  $i, j, y, z$

$k = x[(l+r) \div 2][0]$

$i = l$

$j = r$

повторять

пока  $x[i][0] < k$

нч

$i = i + 1$

кц



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№5 (продолжение).

ПОКА  $x[j][0] > k$

НЦ

$j = j - 1$

КЦ

ЕСЛИ  $i < j$  ТО

$y = x[i][0]$

$z = x[i][1]$

$x[i][0] = x[j][0]$

$x[i][1] = x[j][1]$

$x[j][0] = y$

$x[j][1] = z$

$i = i + 1$

$j = j - 1$

ИНАЧЕ

ЕСЛИ  $i = j$  ТО

$i = i + 1$

$j = j - 1$

ВСЁ

ВСЁ

ДО  $i > j$

ЕСЛИ  $i < j$  ТО

$sort(x, i, j)$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



✓5 (продолжение)

ЕСЛИ  $i < r$  ТО  
 $sort(x, i, r)$

КОН

✓4.

алг ЛАБИРИНТ

НАЧ

ЦЕЛ  $n, cnt$

ВВОД  $n$

ЦЕЛ  $arr[n][n], ans[n \cdot n]$

ДЛЯ  $i$  ОТ 0 ДО  $n-1$

НЦ

ДЛЯ  $j$  ОТ 0 ДО  $n-1$

НЦ

ВВОД  $arr[i][j]$

КЦ

КЦ

$cnt = 0$

ЛОГ  $f = \text{ИСТИНА}$

ДЛЯ  $i$  ОТ  $n-1$  ДО 0 С ШАГОМ  $-2$

НЦ

ЕСЛИ  $f = \text{ИСТИНА}$  ТО

ДЛЯ  $j$  ОТ  $n-1$  ДО 0 С ШАГОМ  $-1$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

нч (продолжение)

нц

Если  $\text{prime}(\text{arr}[i][j]) = \text{истина}$  то $\text{ans}[\text{cnt}] = \text{arr}[i][j]$  $\text{cnt} = \text{cnt} + 1$ 

все

кц

иначе

Для ; от 0 до  $n-1$ 

нц

Если  $\text{prime}(\text{arr}[i][j]) = \text{истина}$  то $\text{ans}[\text{cnt}] = \text{arr}[i][j]$  $\text{cnt} = \text{cnt} + 1$ 

все

кц

все

Если  $f = \text{истина}$  то $f = \text{ложь}$ 

иначе

 $f = \text{истина}$ 

все

кц

Вывод "Количество собранных шмелей:"

Вывод  $\text{cnt}$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

нч (продолжение)

Вывод "Собранные шипы:"

Для  $i$  от 0 до  $\text{cnt}-1$ 

нц

Вывод  $\text{ans}[i]$ 

кц

кон

алг prime (арг цел  $x$ )

нач

Если  $(x < 2)$  или  $(x > 2$  и  $x \bmod 2 = 0)$  то

возврат ложь // как return

все

Для  $i$  от 3 до цел( $\sqrt{x}$ ) с шагом 2

нц

Если  $x \bmod i = 0$  то

возврат ложь

все

кц

возврат истина

кон

И в условии и в рисунке к примеру даны разные направления, возьмем то, которое указано на рисунке.

# Олимпиада школьников «Надежда энергетики»

79F-01  
Дистанционно, с использованием ВКС

№ группы

Место проведения

V057-80

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73991

ФАМИЛИЯ ИНГЕРОЙНЕН

ИМЯ АЛЕКСЕЙ

ОТЧЕСТВО АНДРЕЕВИЧ

Дата рождения 11.11.2005

Класс: 9

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.

V057-80



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

## Задача 1

число в шестеричной системе счисления  
 В цикле от 0 до 5 включительно произойдет  $x, y, z$  ( $\text{for } x \text{ in range}(0, 6):$   
 $\text{for } y \text{ in range}(0, 6):$   
 $\text{for } z \text{ in range}(0, 6):$ )  
 где каждому числу  $A$  ( $\text{str}(20) + \text{str}(x) + \text{str}(y) + \text{str}(z) + \text{str}(20) + \text{str}(y) + \text{str}(x) + \text{str}(20) + \text{str}(z) + \text{str}(y) + \text{str}(x)$ )  
 все это в целочисленном типе ( $\text{int}(A)$ ),  
 затем второе число  $B$  ( $\text{int}(\text{str}(y) + \text{str}(z) + \text{str}(x))$ ) и проверим  
 делимость  $A \bmod B$ . Если она равна 0, то эта тройка подходит,  
 иначе не подходит. (if  $A \bmod B = 0$ :  $x, y, z$  подходит)

## Задача 2

Сначала переведем число  $A$  в 10-ю систему счисления. Введем переменную  $i$   
 и переменную  $B$ , пойдём по числу справа (первый будет разряд  
 десятков), первым - единицу. Число (покажем из разрядов)  
 Будем брать  $i$  разряд и увеличивать  $B$ , если ставим цифру  
 из возможных элементов числа (1-1, 2-2 ... A-10, B-11, C-12, D-13, E-14, F-15)  
~~for i in range(len(A)-1, -1, -1):~~

## Задача 3

Переведем число  $A$  в 10-ю систему счисления. Произойдет в цикле  
 по каждому элементу и будем добавлять в  $B$   $x \cdot 16^z$ , где  
 $x$  это составленный элемент (0-0, 1-1, ... A-10, B-11, ... F-15) и  $z$  это  
 разряд считая справа от 0.  $z=0, B=0, x=0$   
 приближенной код:  $\text{for } i \text{ in range}(0, \text{len}(A))$ :  
 $B = B + 16^i \cdot \text{int}(A[-i-1])$

теперь переведем число по сс. в латиницу.  
 Сделаем в  $C$  строку  $(\text{str}(\text{int}(B)) + \text{str}(\text{int}(B)))$ .  
 + " " где  $B$  это разряд из  $B$  считая от  
 правой стороны.

$B = \text{str}(A), C = \text{str}(B)$   
 $\text{for } i \text{ in range}(\text{len}(B))$ :  
 $C = C + (\text{str}(\text{int}(B[-i-1])) + \text{str}(\text{int}(B[-i-1])))$

исходное число выписано в  $C$

$B = \text{str}(A)$   
 $\text{for } i \text{ in range}(\text{len}(B))$ :  
 $B = B + 16^i \cdot \text{int}(B[-i-1])$   
 $C = C + \text{str}(\text{int}(B[-i-1])) + \text{str}(\text{int}(B[-i-1]))$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Задача 5

$$\begin{array}{cccc}
 A \rightarrow C & A \& B \rightarrow D & \neg B \rightarrow E & C, D, E = 0 \\
 \text{"} & \text{"} & \text{"} & & \\
 1 & 1 & 1 & & 
 \end{array}$$

1)  $A \rightarrow C \Rightarrow A=0$   
т.е.  $A$   $10 \times 6$

A	C	
0	0	1
0	1	1
1	0	0
1	1	1

2)  $A \& B \rightarrow D$

A	B	A&B	D	A&B → D
0	0	0	0	1
0	1	0	0	1
1	0	0	0	1
1	1	1	0	0
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

тогда что  
не подходит  
A-ложь, а не истина  
и наоборот - истина, т.е. 1  
не подходит  
D-ложь

3)  $\neg B \rightarrow E$

B	$\neg B$	E	$\neg B \rightarrow E$
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1

не подходит, истинное  
ложь  
не подходит, E-истина  
не подходит, E-истина

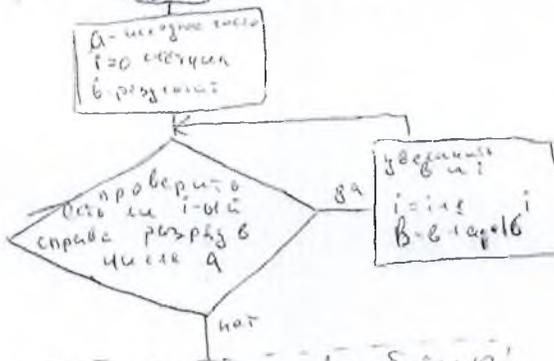
из 3 следует, что B истина,  
других вариантов в нет



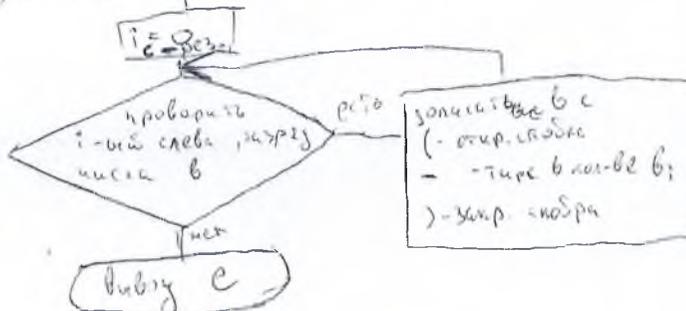
Ответ: A-ложь, B-истина.

Задача 3 (блок-схема)

1) проверка  $10 \leq a \leq 100$



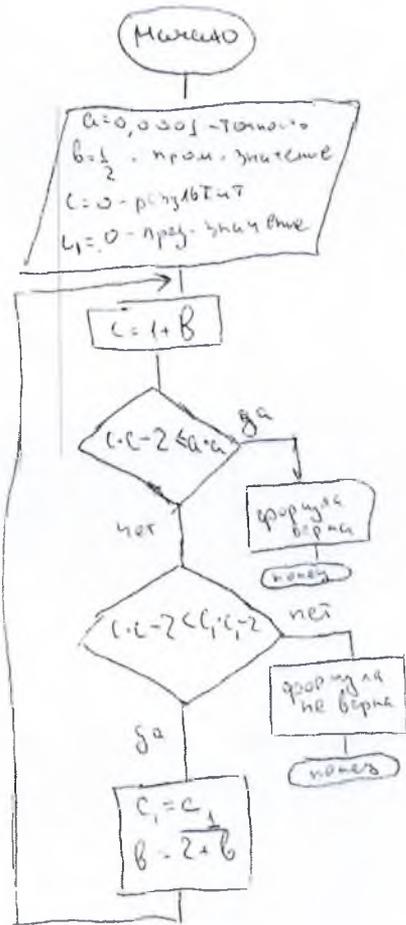
2) проверка из  $10 \leq a \leq 100$  в диапазоне





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

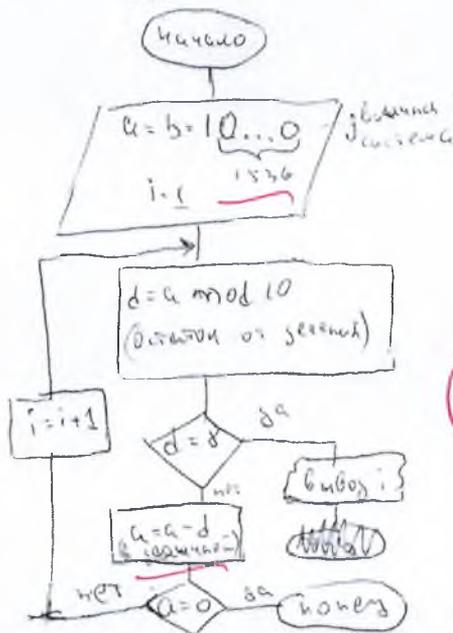
Задача 4



(+)

Задача 2

$$64^{256} = 8^{512} = \left(\frac{1}{2}\right)^{512} = 2^{-512} = \frac{10^{i-1} \cdot 2}{1536} = 10^i \cdot a_i + 10^{i-1} \cdot a_{i-1} + \dots$$



(+)

Олимпиада школьников «Надежда энергетики»

I11F03 Дистанционно,  
с использованием ВКС  
№ группы Место проведения

GR14-48  
шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 43111

ФАМИЛИЯ Крупцов  
ИМЯ ИВАН  
ОТЧЕСТВО Игоревич

Дата рождения 11.09.2003

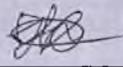
Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.

из 04



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

① Импликация равна нулю, если из правды следует ложь.  
~~Анализ этой функции не нужен.~~

$A \rightarrow C = 1; C = 0; \text{значит, } A = 0$

(+)

$A \wedge B \rightarrow D = 1; D = 0; \text{значит, } A \wedge B = 0; B \text{ либо } 0 \text{ или } 1.$

$\bar{B} \rightarrow E = 1; E = 0; \text{значит, } \bar{B} = 0; B = 1.$

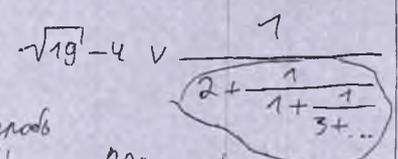
Ответ:  $A = 0, B = 1$

② Для удобства на каком-то этапе будем ставить множителем дробь  $\sqrt{19} - 4$ . Так же, ввиду непрерывности операций с вещественными числами в ЭВМ будем рассчитывать результат в обнн. дробях. Здесь и далее машина индексирована с 1

```

val = sqrt(19) - 4
mas = [2, 1, 3, 1, 2, 8]; mas_pos = 1
nom = 2; denom = 1
ПОКА abs(val - denom / nom) > 0,0001:
    ИЛИ ЕСЛИ mas_pos > 6:
        mas_pos = 1
    ИНАЧЕ:
        mas_pos += 1
    nom, denom = сложить (nom, denom, 1, mas[mas_pos])

```



nom и denom относятся к этой дроби в знаменателе! (числ. и знам.)

(+)

nom, denom = сложить (nom, denom, 1, mas[mas\_pos])  
кц  
Вывод "утверждение верно!"

Опишем алгоритм "сложить", используемый для сложения дробей  $\frac{a}{b}$  и  $\frac{c}{d}$  ( $\frac{a}{b} + \frac{c}{d} = \frac{a \cdot \text{НОК}(b,d) + c \cdot \text{НОК}(b,d)}{\text{НОК}(b,d)}$ )

```

алг add(a, b, c, d):
    new_denom = НОК(b, d)
    вернуть (a * new_denom / b + c * new_denom / d, new_denom)

```

Опишем НОК и НОД:  
алг НОК(a, b):  
 вернуть (a \* b / НОД)

алг НОД(a, b):  
 если b == 0 вернуть(a)  
 ИНАЧЕ вернуть (НОД(b, a % b))

← Алг. Евклида



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

И) Так как нам нужно пройти из левого нижнего угла в правый верхний, мы не затрагивали вершнюю и нижнюю строки. Если в ~~лабиринте~~ <sup>лабиринте</sup> ~~матрице~~ <sup>матрице</sup>  $n$ , ~~длина~~ <sup>длина</sup> ~~пути~~ <sup>пути</sup> считать, что индекс строки с центральным ~~кроссовером~~ <sup>кроссовером</sup> лабиринта равен  $n // 2$  (целая часть).

Пройдем по ~~пути~~ <sup>пути</sup> ~~матрице~~ <sup>матрице</sup> и найдем наибольшее число  $z$  (0 < z):  
 $mas\_num = 0$  ~~и~~ <sup>и</sup> ~~найдем~~ <sup>найдем</sup> ~~наибольшее~~ <sup>наибольшее</sup> ~~число~~ <sup>число</sup>  $z$  (0 < z):  
 $max\_num = 0$  ~~и~~ <sup>и</sup> ~~найдем~~ <sup>найдем</sup> ~~наибольшее~~ <sup>наибольшее</sup> ~~число~~ <sup>число</sup>  $z$  (0 < z):

Для  $i$  от  $n // 2$  до  $0$ , шаг  $-1$ :

НУ если  $mas[i][0] > max\_num$ :  
 $max\_num = mas[i][0]$

кц

Для  $j$  от  $1$  до  $n$ :

НУ если  $mas[i][j] > max\_num$ :  
 $max\_num = mas[i][j]$

кц

Для  $i$  от  $n // 2$  до  $n$ , шаг  $-1$ :

НУ если  $mas[i][n] > max\_num$ :  
 $max\_num = mas[i][n]$

кц

Теперь нам нужно ~~найти~~ <sup>найти</sup> ~~простые~~ <sup>простые</sup> ~~числа~~ <sup>числа</sup> до  $max\_num$ .  
 $primes[true, true, \dots, true]$   
 $primes[i] - \text{простое ли } i?$

~~длина~~ <sup>длина</sup> ~~матрицы~~ <sup>матрицы</sup>  $cur = 0$

Пока  $d \cdot d \leq n$ :

НУ  $cur = d \cdot d$

Пока  $cur < max\_num$ :  
НУ  $primes[cur] = false$   
 $cur += d$

кц  
 $d += 1$

В  $primes$  теперь просты все числа на ~~интервале~~ <sup>интервале</sup> ~~от 0 до~~ <sup>от 0 до</sup> ~~max\\_num~~ <sup>max\\_num</sup>.

Теперь, используя ~~одно~~ <sup>одно</sup> ~~число~~ <sup>число</sup>  $z$ , в теле каждого цикла проверяем, верно ли  $primes[mas[i][j] - z]$ . Если верно, вывели координаты  $i$  и  $j$  и ее значение.



Олимпиада школьников «Надежда энергетики»

Вариант: 43111

ШИФР, НЕ ЗАПОЛНЯТЬ! ⇨

GR14-41

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

5) Нам нужно отсортировать массив  $mas$  по значениям  $x$  (пусть  $mas[i][1] = x_i, mas[i][2] = y_i$ ).  
Для этого используем сортировку слиянием:

нужно нам дан массив  $mas$ .  
Еще его функция ~~вернется~~ <sup>не сортирует</sup>, то применим сортировку слиянием.  
Для этого его разделим (на  $[0 : n/2]$ ,  $[n/2 : n]$  чел-функция).  
Получим две функции ~~внутренние~~, удалим все элементы из  $mas$ .  
Далее сортируем полученные массивы в массив:  
~~в~~  $mas$  записываем меньший из крайних ~~элементов~~ <sup>элементов</sup> ~~двух массивов~~ и удалим этот элемент из ~~каждого~~ <sup>каждого</sup> массива.  
Если одна из половинок ~~кратна~~ <sup>больше</sup> добавим все элементы ~~мелочной~~ <sup>мелочной</sup> половины в конец  $mas$ .  
Если длины равны ~~сортим~~, ~~каждый~~  $mas$  из ~~каждого~~ <sup>каждого</sup> элемента ~~отсортируем~~.

~~Важно~~ ВАЖНО: в сортировке мы сравниваем первые элементы пар значений  $m$ . Сортировка

$[ [0, 2], [2, 1], [1, 5] ]$  мы смотрим на первое элемент <sup>(0, 2, 1)</sup> ~~и сортируем~~

Итак, мы получили массив пар  $[x_i, y_i]$ , который отсортирован по  $x$  значениям.

То есть, второе значение пар — последовательное значение ~~исходной~~ <sup>исходной</sup> функции. Теперь пройдем весь массив, на каждом шаге проверяя ~~на~~ <sup>на</sup> ~~равенство~~  $y_i \leq y_{i+1}$ .

для  $i$  от 1 до  $n-1$ :  
if  $y_i > y_{i+1}$ :  $flag = true$

else  $y_i > y_{i+1}$ :  $flag = false$   
вывод "Функция не монотонно возрастает!"  
кц ~~выход~~ <sup>выход</sup> из цикла

если  $flag$ :  
вывод "Функция монотонно возрастает!"





ВНИМАНИЕ! Проверьте только то, что записано с этой стороны листа в рамке справа

③ Пусть робот изначальнo находится в точке [m][0]. Разместим, в каждой ступеньке может находиться робот, кепард в точке [i][j]

1) Если в [i-1][j] нет ступеньки, робот может прийти только из [i-1][j-1] или [i-1][j+1].  
•  $x = i-1; y = j-1, j+1, j+3$   
•  $x \geq i; y = j+3$

2) Если в [i-1][j] нет ступеньки;  
•  $x = i-1; y = j-1, j-2, j-3$   
•  $x \geq i; y = j+3$

3) Если в [i-1, i-2 или i-3][j] нет ступеньки;  
•  $x \geq i-3; y = j+1$

4) Если в [i-1, i-2 или i-3][j] нет ступеньки;  
•  $x \geq i-3; y = j-1$

Теперь порядок ячеек used с помощью массива, во всех ячейках со ступеньками по умолчанию стоит false

Далее формируем ответ из ячейки H.

Используем алгоритм ОБХОДА: \* ТРАЕКТОРИИ  
• проверить, в какие клетки можно прийти с помощью разных траекторий  
• добавить в used, вызвать ОБХОД от [i][j] и добавить в used всех достижимых клеток в used  
\* Если все возможные клетки, в которых можно оказаться, в used, перейти к следующей траектории

Если по окончании алгоритма мы не побывали в клетке K, то в неё обратиться нельзя. Если же used по координатам K все-таки true, значит, во верхней левой клетке обратиться можно

\* ЗА ОДИН ХОД РОБОТА, и.е. НА КАКИХ СТУПЕНЬКАХ ЗА ОДИН ТРАЕКТОРИЮ.

# Олимпиада школьников «Надежда энергетики»

I11F02	Дистанционно, с использованием ВКС
№ группы	Место проведения

GR 98-63
----------

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

Вариант № 73111

ФАМИЛИЯ Кутасин

ИМЯ Владимир

ОТЧЕСТВО Алексеевич

Дата рождения 01.12.2003

Класс: 11

Предмет информатика

Этап: Заключительный

Работа выполнена на 10 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:

Кутасин

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N 1

$$\begin{cases} A \rightarrow C = 1 \\ A \& B \rightarrow D = 1 \\ \neg B \rightarrow E \end{cases}$$

$$C = 0, D = 0, E = 0$$

$$A = ?, B = ?$$

$$\begin{cases} A \rightarrow 0 = 1 \\ A \& B \rightarrow 0 = 1 \\ \neg B \rightarrow 0 = 1 \end{cases}$$

Из первого ур-ния следует, что  $A = 0$ ; ⊕

Из ~~второго~~<sup>третьего</sup> ур-ния следует, что  $B = 1$ ;

При этом ~~из~~ второе ур-ние также верно ( $0 \& 1 \rightarrow 0 = 1$ ).

↓  
Ответ:  $A = 0, B = 1$ .

Истина?  
Ложь?



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N 2

$$\Delta = 0,0001$$

ЭВМ с доской способны только получать  $\sqrt{15}$ .

Значит нам следует вычислить значение вычисления правой части ур-ния. Я вижу два разряда между:

1) Разрешаю итер. кол-во итераций чисел (2, 1, 3, 1, 2, 6) и при недостаточности точности увелич. итер. кол-во:

алг Задача - 2

нач  
цел i = 2  
вещ сир, prev  
вещ e = 0,0001  
сир = Подзем(i)

~~для i от 1 до 6~~

ПОКА ИСТИНА

нц

prev = сир

~~для i от 1 до 6~~

сир = Подзем(i)

i = i + 1

ЕСЛИ abs(сир - prev) < e ТО

ВЫХОД. ИЗ ЦИКЛА

ВСЁ

кц

ЕСЛИ abs(sqrt(19) - сир) > e ТО

ВЫВОД ("ЛОЖЬ")

ИНАЧЕ

ВЫВОД ("ИСТИНА")

ВСЁ

кон

Асимптотика такого решения  $O(P^2)$

2) Найти ответ для большего периода и сравнить с  $\sqrt{15}$ :

В ходе вычисления только подзем итер. кол-во, достаточно большое для

сир = Подзем(1000000)

алг Подзем (арт цел p)

нач

вещ res = 0

для i от 1 до p

нц

res = 1 / (2 + 1 / (1 + 1 / (3 + 1 / (1 + 1 / (2 + 1 / (6 + res))))))

кц

ВЕРНУТЬ res + 4

кон

$$\text{abs}(x) = |x|$$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Асимптотика такого решения  $O(p)$ , но мы не можем точно сказать, правильное ли вычислено значение. Самый оптимальный вариант это первое решение, с увелич. массы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N4

Такой образ ~~111111~~ матрицы ~~составить~~ <sup>числами 3</sup> ~~невозможно~~ <sup>можно</sup>  
Проверку на простоту можно сделать ~~с помощью~~ <sup>2</sup> способами:

1) Проверка всех делителей самого числа до корня:  
 $O(N^2 \sqrt{N})$

2) Или воспользоваться решеткой Эратосфена  $O(N^2 / \log N + M \log \log M)$   
где M - это максимальное число.

Копию второй алгоритма  
или Задача - 4

Нач

```
цел prime [M]
prime[i] = ложь
цел i = 2, j
```

```
Пока i <= M
```

```
нц
  Если prime[i] = истина
```

```
  j = i + i
```

```
  Пока j <= M
```

```
  нц prime[j] = ложь
```

```
  j = j + i
```

```
нц
```

```
всё
```

```
кц
```

```
цел ans = 0
```

```
цел x = 1, y = N
```

```
Пока истина
```

```
нц
```

```
  Если x = N и y = 1 то
```

```
    Выход из цикла
```

```
всё
```

```
  Если (N - y) mod 2 = 1 то
```

```
    y = y - 1
```

```
иначе
```

```
  Если (N - y) mod 4 = 0 то
```

```
    Если x = N то
```

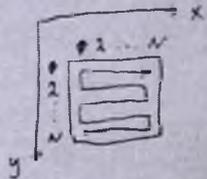
```
      y = y - 1
```

```
иначе
```

```
  x = x + 1
```

```
всё
```

```
цел N, вввод(N)
цел arr [M][N]
цел M = 0
для a от 1 до N
  нц
    для b от 1 до N
      вввод(arr[a][b])
      Если arr[a][b] > M то
        M = arr[a][b]
всё
```



```
1: для a от 2 до M
нц
  prime[a] = истина
кц
```

```
2: Если prime[arr[x][y]] = истина то
  ans = ans + 1
всё
```

```
3: Если N != 1 то
  2
всё
```





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



ИНАЧЕ

ЕСЛИ  $x = 1$  ТО

$$y = y - 1$$

ИНАЧЕ

$$x = x - 1$$

ВСЁ



ВСЁ

~~ВСЁ~~ ②

КЦ

Вывод (ans)

КОН



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N 5

Возьмем все точки, напомним как пары  $(x, y)$ <sup>массив</sup>, отсортируем по  $x$ <sup>(возраст.)</sup> и проверим  $y$ . Сортировка - QSort.

~~Пара  $(x, y)$~~

Структура Пара:

вещ  $x, y$

алг Задача - 5

нццел  $N$ , вввод ( $N$ )

Пара  $a [N]$

для  $i$  от 1 до  $N$ .

нц вввод ( $a[i].x$ )

кц для  $i$  от 1 до  $N$

нц вввод ( $a[i].y$ )

кц QSort ( $a, 1, N$ )

лог  $res = \text{ИСТИНА}$

для  $i$  от 2 до  $N$

нц ~~ЕСЛИ  $(a[i].y \neq a[i-1].y)$~~

ЕСЛИ  $(a[i].x = a[i-1].x)$  ТО

ЕСЛИ  $a[i].y \neq a[i-1].y$  ТО

ВЫВОД ("Ошибка")  
ВСЕ Выход-из-программы

ИНАЧЕ ЕСЛИ  $a[i].y \leq a[i-1].y$  ТО

$res = \text{ЛОЖЬ}$   
ВСЕ Выход-из-цикла

кц

ЕСЛИ  $res = \text{ИСТИНА}$  ТО

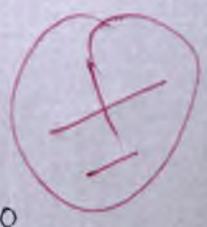
ВЫВОД ("ИСТИНА")

ИНАЧЕ

ВЫВОД ("ЛОЖЬ")

ВСЕ

КОН



*Неправильно*



ВНИМАНИЕ! Посеребруется только то, что записано  
с этой стороны листа в рамке справа

алг QSort (арг рез Пара  $a[N]$ , арг цел first, арг цел last)

нлч

цел  $i = \text{first}$ ,  $j = \text{last}$

Пара  $k = a[(i+j) \text{div } 2]$ ,  $y$

повторять

пока  $a[i].x < k.x$

нц

$i = i + 1$

кц

пока  $a[j].x > k.x$

нц

$j = j - 1$

кц

ЕСЛИ  $i \leq j$  ТО

$y = a[i]$

$a[i] = a[j]$

$a[j] = y$

$i = i + 1$

$j = j - 1$

ВСЕ

пока  $i \leq j$

ЕСЛИ  $j > \text{first}$  ТО

QSort( $a$ , first,  $j$ )

ВСЕ

ЕСЛИ  $i < \text{last}$  ТО

QSort( $a$ ,  $i$ , last)

ВСЕ

кон

Асимптотика рекур.:  $O(N \log N)$ .

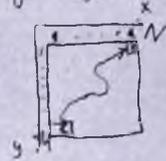


ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3

Для решения данной задачи поступим след образом  
Заведём ~~матрицу~~ <sup>матрицу used</sup>, в которой будем хранить  
информацию о том, посетил ли она. <sup>для каждой клетки</sup>

Далее закончим поиск в глубинку из нач. клетки.  
~~Итак~~ Из used узнаем ответ на задачу. Асимптотическое  
решение:  $O(N \cdot M)$ .



```

int int Задача - 3
int int M, N ВВОД(M) ВВОД(N)
лог pole[M][N], used[M][N]
для y от 1 до M
нц

```

P.S: 1 = ИСТИНА  
0 = ЛОЖЬ

```

    для x от 1 до N
    нц
        симв ch
        ВВОД(ch)
        ЕСЛИ ch = '→' ТО
            pole[y][x] = 1
        ИНАЧЕ
            pole[y][x] = 0
        ВСЕ
        used[y][x] = 0
    кц

```



```

① > кц
    dfs (pole, down, used, M, N)
    ЕСЛИ used[1][N] = 1 ТО
        ВЫВОД ("Смоет")
    ИНАЧЕ
        ВЫВОД ("Не смоет")
    ВСЕ

```

КОН  
Массивы pole и used будут передаваться в dfs по ссылке на ячейки памяти (таким образом, мы сможем избежать лишнего копирования и легко изменять значения в used)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

0  ЦЕЛ down [M][N]
    для j от M до 1
    иц
      для x от 1 до N
      иц
        ЕСЛИ pole[y][x] = 1 ТО
          down[y][x] = y
        ИНАЧЕ
          ЕСЛИ y > M ТО
            down[y][x] = down[y-1][x]
          ИНАЧЕ
            down[y][x] = -1
        ВСЕ
      ВСЕ
    КЦ
  КЦ
  down[y][x] - путь назад из (y, x).

```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

алг dfs (arg ~~списка~~ ~~лог~~ pole[N], arg ~~списка~~ ~~лог~~ used[N], ~~цел~~ x, ~~цел~~ y,  
arg ~~списка~~ ~~цел~~ down[N], arg ~~цел~~ ~~цел~~)

НАЧ

used[y][x] = 1

// ←

ЕСЛИ  $y-1 \geq 1$  И  $x-3 \geq 1$  ТО

ЕСЛИ ~~down[y][x-3] = 1~~ ~~TO~~

ЕСЛИ ~~used[~~y~~][~~x-3~~] = 0~~ И ~~pole[y-1][x] = 0~~ ~~TO~~

dfs (pole, used, ~~x-3~~, ~~y~~)  
down,

BCE  
BCE

// ↑

ЕСЛИ  $y-3 \geq 1$  И  $x-1 \geq 1$  ТО

ЕСЛИ ~~down[y-2][x-1] = 1~~ ~~TO~~

ЕСЛИ ~~used[~~y-2~~][~~x-1~~] = 0~~ И ~~pole[y-1][x] = 0~~ И ~~pole[y-2][x] = 0~~ ~~TO~~

~~pole[y-3][x] = 0~~ И ~~pole[y-2][x-1] = 0~~ ~~TO~~

dfs (pole, used, ~~x-1~~, ~~y-2~~)  
down,

BCE  
BCE

// ↗

ЕСЛИ  $y-3 \geq 1$  И  $x+1 \leq N$  ТО

ЕСЛИ ~~down[y-2][x+1] = 1~~ ~~TO~~

ЕСЛИ ~~used[~~y-2~~][~~x+1~~] = 0~~ И ~~pole[y-1][x] = 0~~ И ~~pole[y-2][x] = 0~~ ~~TO~~

~~pole[y-3][x] = 0~~ И ~~pole[y-2][x+1] = 0~~ ~~TO~~

dfs (pole, used, ~~x+1~~, ~~y-2~~)  
down,

BCE  
BCE

// ↘

ЕСЛИ  $y-1 \geq 1$  И  $x+3 \leq N$  ТО

ЕСЛИ ~~down[y][x+3] = 1~~ ~~TO~~

ЕСЛИ ~~used[~~y~~][~~x+3~~] = 0~~ И ~~pole[y-1][x] = 0~~ И ~~pole[y][x+1] = 0~~ ~~TO~~

dfs (pole, used, ~~x+3~~, ~~y~~)  
down,

BCE  
BCE

КОН

P.s.: N знаем из размера pole или used.

# Олимпиада школьников «Надежда энергетики»

СДРОЛ	Александрово, с. Использование ВКС
№ группы	Место проведения

QV 19-63
----------

Не распечатать.  
Занимается  
ответственным  
работником

Вариант № 33101

цифр

ФАМИЛИЯ Маларевич

ИМЯ Мария

ОТЧЕСТВО Александровна

Дата рождения 21.02.2007

Класс: 10

Предмет История

Этап: Заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 21.02.2021

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Прозвучит только то, что записано с этой стороны листа в рамках стрел

$$\begin{cases}
 A \rightarrow C = 1 \\
 A \& B \rightarrow D = 1 \\
 B \rightarrow E = 1
 \end{cases}
 \quad
 \begin{matrix}
 \text{знач.:} \\
 C=0, D=0, E=0 \\
 A=? \\
 B=?
 \end{matrix}$$

Решение:  
 1) По к.  $C=0$ , а  $A \rightarrow C = 1$ , а может быть равно только 0 ( $0 \rightarrow 0 = 1$ )  
 2) Подставим известные значения

$A, D$  в старое утверждение:  $0 \& B \rightarrow 0 = 1$ . Оно верно при любых  $B$ .

3) Подставим известные значения в второе утверждение:  $0 \rightarrow 0 = 1$   
 $B=0 \Rightarrow B=1$  ( $0 \rightarrow 0 = 1$ )

Ответ:  $A=0, B=1$ . (1)

10) Создадим двумерный массив  $tabl$ , в котором  $N$  строк, каждый из которых является массивом из строк  $M$ , все элементы которого равны 0. Далее создаем в блоке  $i$  и  $j$  в другой цикл, в первом из которых перебираем  $i$  и изначально равна  $N-1$  и с каждым вхождением в цикл уменьшаем на 1 до тех пор, пока не достигнем значения  $-1$  (т.е. для  $i=0$  цикл еще  $i$  раз пройдет, а для  $i=-1$  - нет), а во втором - перебираем  $j$ , изначально равна 0 и увеличиваем на 1 до тех пор, пока не достигнет значения  $M$  ( $j=M-1$  - цикл выполнится,  $j=M$  - нет). При каждом вхождении во второй цикл мы рассматриваем клетку  $tabl[i][j]$ . Проверим существование клетки  $tabl[i+1][j-3]$  (т.е. проверка, что  $i+1 < N, j-3 \geq 0$ ). Если эта клетка существует, то проверим наличие колотки в клетках  $tabl[i][j-3], tabl[i][j-2], tabl[i][j-1]$ . Если какая-то одна из них содержит колотку, то мы делаем значение  $tabl[i][j] = 1$  и переходим к проверке следующей клетки. Иначе проверяем существование клетки  $tabl[i+1][j+3]$  (т.е.  $i+1 < N, j+3 < M$ ), и, если она существует, равно ли ее значение 1. Если оба условия выполнены, то проверяем наличие колотки в клетках  $tabl[i][j+3], tabl[i][j+2], tabl[i][j+1]$ , и действуем аналогично предыдущей проверке. Если  $tabl[i][j+1]$  все равно = 0, то осуществляем проверку существования, и, если клетка существует и ее знак = 1, то, как и в предыдущих клетках, проверяем наличие колотки в клетках  $tabl[i+2][j+1], tabl[i+1][j+1]$  и  $tabl[i][j+1]$ , и действуем аналогично предыдущей проверке. Иначе, наконец, если  $tabl[i][j]$  все равно 0, проверяем существование дальнейшей клеточки  $tabl[i+3][j-1]$  ( $i+3 < N, j-1 \geq 0$ ), и наличие колотки в клетках  $tabl[i+2][j-1], tabl[i+1][j-1], tabl[i][j-1]$ . Если на в одной из них нет колотки, то  $tabl[i][j] = 1$ .

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке стрелой

Идея вычисления датских чисел проверки заключается в том, что если  $tbl[0][M-1]$ . Если оно равно 1, то до числа  $K$  можно добраться от числа  $K$ , иначе - нельзя.

\* проверка существования колонки на данной клетке проводится с помощью двумерной таблицы  $M \times N$ , а не с помощью массива  $tbl$ .

Код:

#  $M, N$  - известные нам числа,  $t$  - двумерная таблица  $M \times N$ .

```
tbl = [[0 for j in range(M)] for i in range(N)]
```

```
tbl[N-1][0] = 1
```

```
for i in range(N-1, -1, -1):
```

```
    for j in range(0, M, 1):
```

```
        if i+1 < N and j-3 >= 0:
```

```
            if tbl[i+1][j-3] == 1:
```

```
                a = t[i][j-3]
```

```
                b = t[i][j-2]
```

```
                c = t[i][j-1]
```

```
                if a != "x" and b != "x" and c != "x":
```

```
                    tbl[i][j] = 1
```

```
                    continue
```

```
        if i+1 < N and j+3 < M:
```

```
            if tbl[i+1][j+3] == 1:
```

```
                a = t[i][j+3]
```

```
                b = t[i][j+2]
```

```
                c = t[i][j+1]
```

```
                if a != "x" and b != "x" and c != "x":
```

```
                    tbl[i][j] = 1
```

```
                    continue
```

```
        if i+3 < N and j+1 < M:
```

```
            if tbl[i+3][j+1] == 1:
```

```
                a = t[i+3][j+1]
```

```
                b = t[i+2][j+1]
```

```
                c = t[i+1][j+1]
```

```
                if a != "x" and b != "x" and c != "x":
```

```
                    tbl[i][j] = 1
```

```
                    continue
```

```
        if j+3 < N and j-1 >= 0:
```

```
            if tbl[i+3][j-1] == 1:
```

```
                a = t[i+3][j-1]
```

```
                b = t[i+2][j-1]
```

```
                c = t[i+1][j-1]
```

```
                if a != "x" and b != "x" and c != "x":
```

```
                    tbl[i][j] = 1
```

```
if tbl[0][M-1] == 1:
```

```
    print("yes")
```

```
else:
```

```
    print("no")
```





ВНИМАНИЕ! Проставьте голубой за. или черной с этой стороны листа в рамке справа

Пусть  $\{x_i\}_{i=1}^N$  — последовательность действительных чисел, удовлетворяющая условиям  $x_i \in [0, 1]$  и  $x_i + x_{i+1} \leq 1$  для  $i=1, \dots, N-1$ . Пусть  $S = \sum_{i=1}^N x_i$ . Найти максимум функции  $f(x_1, \dots, x_N) = \sum_{i=1}^N x_i^2$  при заданных условиях.

Решение. Рассмотрим функцию  $f(x_1, \dots, x_N) = \sum_{i=1}^N x_i^2$ . Условия задачи можно переписать как  $x_i \geq 0$  и  $x_i + x_{i+1} \leq 1$  для  $i=1, \dots, N-1$ . Максимум функции  $f$  достигается на границе допустимой области, т.е. когда  $x_i + x_{i+1} = 1$  для  $i=1, \dots, N-1$ . Пусть  $x_1 = y_1$ ,  $x_2 = y_2$ ,  $\dots$ ,  $x_N = y_N$ . Тогда  $y_i + y_{i+1} = 1$  для  $i=1, \dots, N-1$ . Если  $N$  нечетно, то  $y_1 = 1, y_2 = 0, y_3 = 1, y_4 = 0, \dots, y_N = 1$ . Если  $N$  четно, то  $y_1 = 1, y_2 = 0, y_3 = 1, y_4 = 0, \dots, y_N = 0$ . В обоих случаях  $f = \sum_{i=1}^N y_i^2 = \lfloor \frac{N+1}{2} \rfloor$ . Следовательно, максимум функции  $f$  равен  $\lfloor \frac{N+1}{2} \rfloor$ .

1111

1111

# Олимпиада школьников «Надежда энергетики»

I11F02	Дистанционно, с использо ванием ВКС
--------	--

№ группы Место проведения

GR 98-11
----------

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73111

шифр

ФАМИЛИЯ Маничев

ИМЯ АЛЕКСЕЙ

ОТЧЕСТВО ПАВЛОВИЧ

Дата рождения 27.07.2003

Класс: 11

Предмет Информатика

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на \_\_\_\_\_ листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

$$\begin{array}{l}
 \text{№1} \\
 \left\{ \begin{array}{l} A \rightarrow C \\ A \cdot B \rightarrow D \\ \bar{B} \rightarrow E \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \bar{A} + C \\ \bar{A} \cdot B + C \\ B + E \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} A + C = 1 \\ \bar{A} + \bar{B} + C = 1 \\ B + E = 1 \\ C = 0 \\ D = 0 \\ E = 0 \end{array} \right. \Leftrightarrow \tau
 \end{array}$$

$$\Leftrightarrow \left\{ \begin{array}{l} \bar{A} + 0 = 1 \\ \bar{A} + \bar{B} + 0 = 1 \\ B + 0 = 1 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} A = 0 \\ B = 1 \end{array} \right. \quad (\pm)$$

Ответ: A - ~~ложь~~ <sup>ложь</sup> B - истина

№5. алг Задача 5

нац

цел  $N, i, j, d, t, c$

ввод  $N$

цел  $a[x] \in N, a[y] \in N$  // координаты по "x" и "y"

// считывание данных

для  $i$  от 1 до  $N$

кц

ввод  $a[x]$

ввод  $a[y]$

и на этом этапе совпадают индексы углов

и далее упорядочу массив  $a[x]$  и сохраню ординаты тех углов

// по возрастанию

\* кц

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

для  $i$  от 1 до  $N-1$

нц  
для  $j$  от  $i+1$  до  $N$

нц  
если  $a[x[i]] > a[x[j]]$  то

// замена "x"

$t = a[x[i]]$

$a[x[i]] = a[x[j]]$

$a[x[j]] = t$

// замена "y"

$c = a[y[i]]$

$a[y[i]] = a[y[j]]$

$a[y[j]] = c$

все

кц

кц  
~~кц~~

// проверка на возраст. ф-ии

для  $i$  от 1 до  $N$   
нц  
если

лог  $d$

$d = \text{Истина}$

для  $i$  от 1 до  $N-1$

нц

если  $a[y[i]] < a[y[i+1]]$  то

вывод "функция монотонно не возрастает"

$d = \text{Ложь}$

все

кц  
~~кц~~

*Handwritten red notes:*  
уменьше  
кц



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

если  $d = \text{ИСТИНА}$  то  
вывод "функция монотонно возрастает"  
все

кОН

НЧ.

алг simple (цел  $a$ ) и проверка на простоту  
наз цел  $i$

если  $a = 1$  то

вернуть ложь

все

для  $i$  от 2 до  $a \div 2$

нч

если  $a \% i = 0$  то

вернуть ~~ложь~~ ложь

все

кч

вернуть истина

основная прог-мма

алг задача 4

наз

цел  $k, x, y, n, i, b[1..n]$

лог  $d$

$d = \text{ИСТИНА}$

ввод  $n$

цел  $a[1..n]$

для  $i$  от 1 до  $n$

нч

для  $j$  от 1 до  $n$

нч

ввод  $a[i, j]$

кч

кч



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

для  $i$  от 1 до  $n \cdot n$

ну

~~все~~  $b[i] = 0$  // ~~каждый~~ массив нулевых имеет размер  $n \cdot n$

ку

$k = 0$  // количество шагов нулевых

// начальная точка

$x = n$

$y = n$

$d = \text{Истина}$  // условие работы цикла

~~иначе~~

ПОКА  $d = \text{Истина}$

ну

// движется влево

пока  $x \neq 1$

ну

если  $\text{simple}(a[y, x]) = \text{Истина}$  ТО

$b[k] = a[y, x]$

$k = k + 1$

все

$x = x - 1$

ку

если  $y \neq 1$  или  $x \neq 1$  ТО проверка тут, т.к. заверш. в лев. верх.

// 2 шага вверх

~~и~~ для  $i$  от 1 до 2

ну

если  $\text{simple}(a[y, x]) = \text{Истина}$  ТО

$b[k] = a[y, x]$

$k = k + 1$

все

$y = y - 1$

ку

Проверка



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



«двигаемая вправо»  
пока  $x \neq \text{нич}$

нч

если  $\text{simple}(a[y, x]) = \text{ИСТИНА}$  то

$b[k] = a[y, x]$

$k = k + 1$

все

$x = x + 1$

кч

«двигалась вверх»  
для  $i$  от 1 до 2

нч

если  $\text{simple}(a[y, x]) = \text{ИСТИНА}$  то

$b[k] = a[y, x]$

$k = k + 1$

все

$y = y - 1$

кч

ИНАЧЕ

$d = \text{ЛОЖЬ}$

все // if  $y \neq 0 \dots$

кч // while

для  $i$  от 1 до  $k$

нч

вывод  $b[i]$

кч

⊛: если  $\text{simple}(a[y, x]) = \text{ИСТИНА}$   
 $b[k] = a[y, x]$   
 $k = k + 1$   
все

КОН условия

// В задаче и рисунке есть разногласия. Задача сделана в соответствии с рисунком. Если все подразумевалось решение в соответствии с текстом задачи, то начальные координаты будут  $y = n$ ,  $x = 0$  и иной порядок операций в «нч»: «вверх, вправо, вверх влево». Проверка будет после второго «вверх».

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№2.

алг Задача 2

нач

~~цел my, my\_past, i~~

вещ my, my\_past, a

цел i, k[6]

k = [2, 1, 3, 1, 2, 8]

~~докв my~~

my = 4

my\_past = 0

i = 1

пока  $|my - my\_past| > 0,0001$ 

нц

my\_past = my

my = my + 1/k[i]

если i = 6 то

i = 1

иначе

i = i + 1

все

кцн // while

a = корень квадратный из  $(19) * 10000 / 10000$ 

ц если a = my то

вывод "формула справедлива"

иначе

вывод "формула не справедлива"

вс

кон

// my - текущий результат. my\_past - результат действия назад.

// a - табличное значение корня из 19-ти до знака после запятой 500 знаков посл. запятой

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

алг fall (a, steps, y, x, m, k)
нач лог f
f = ИСТИНА
ПОКА f = ИСТИНА и падаем вниз
нц
  если a[y, x] = 1 и ушли на ступеньку
  если steps[y, x] = k + 1 то
    вернуть steps
  все
  y = y + 1 // проходим падаю
  если y >= m то // ушли вниз за границу
    вернуть steps
  все
кон

```

```

алг Задача3
цел n, m, k, i, j, a[m][n], steps[m][n], y, x
всё ввож n
мхк ввож m
для i от 1 до n
  для j от 1 до m
    асij ввож a[i, j]
для i от 1 до n
  для j от 1 до m
    steps[i, j] = n * m * 2 и там указательно определено
кц
кц
// будем использовать волновой алгоритм
лог d = ИСТИНА
k = 1 // номер текущего шага
steps[m, 1] = k // нижний левый угол
ПОКА d = ИСТИНА
нц // найдём где был сделан первый шаг

```

±

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

для  $y$  от 1 до  $m$

нужно для  $x$  от 1 до  $n$

нужно

если  $steps[y, x] = k$  то

// пробуем сделать каждый из шагов

// up right

если  $y > 3$  и  $x < n$  и  $a[y-1, x] + a[y-2, x] + a[y-3, x] = 0$  то

$steps = \text{fall}(a, steps, y-3, x+1, m, k)$

все

// up left

если  $y > 3$  и  $x > 1$  и  $a[y-1, x] + a[y-2, x] + a[y-3, x] = 0$  то

$steps = \text{fall}(a, steps, y-3, x-1, m, k)$

все

// right up

если  $y > 1$  и  $x < n-3$  и  $a[y-1, x] = 0$  то

$steps = \text{fall}(a, steps, y-1, x+3, m, k)$

все

// right left

если  $y > 1$  и  $x > 3$  и  $a[y-1, x] = 0$  то

$steps = \text{fall}(a, steps, y-1, x-3, m, k)$

все

все

$k = k + 1$  // следующий шаг

$k_y$

$k_x$

если  $steps[1, n] = k$  то // пришли в кон. точку

вывод "дойти можно"

$d = \text{True}$  // да

все

ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



28 если  $k=2n+1$  то

$d = \text{False}$

вывод "выхода нет"

все  
конец  
кон

## Олимпиада школьников «Надежда энергетики»

19701 Дистанционно,  
с использованием ВКС

№ группы

Место проведения

VO 57-47

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73991

ФАМИЛИЯ Мартин

ИМЯ Никита

ОТЧЕСТВО Дмитриевич

Дата рождения 27.03.05

Класс: 9

Предмет Информатика

Этап: Заключительный

Работа выполнена на 4 листах

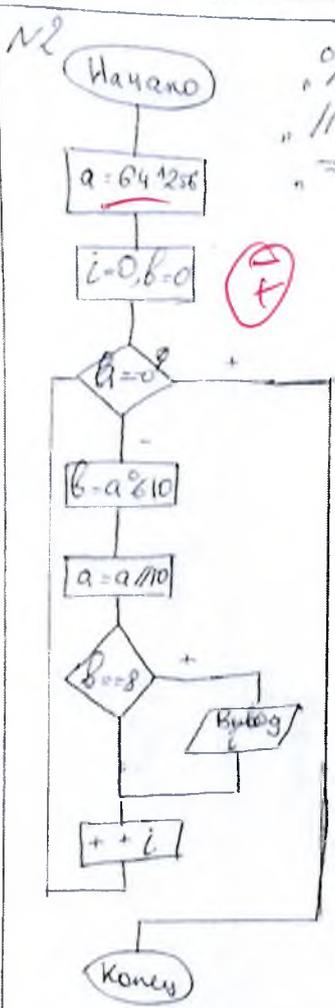
Дата выполнения работы: 28.02.21  
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



$\%$  - остаток от деления  
 $//$  - целочисленное деление  
 $==$  - проверка равенства

№5

Построим таблицу истинности

A	B	C	D	E	$A \rightarrow C$	$A \& B \rightarrow D$	$B \rightarrow D$
0	1	0	0	0	1	1	1

Так  $A \rightarrow C = 1$ , где  $C = 0 \Rightarrow$   
 $\Rightarrow A$  может быть равен только 0.  
 Так  $\bar{B} \rightarrow D = 1$ , где  $D = 0 \Rightarrow$   
 $\Rightarrow \bar{B}$  может быть равен только 0  $\Rightarrow$   
 $\Rightarrow B = 1$

Ответ: A=0; B=1

№4

1) ~~Округлим число  $52^7$  до точности 0,0001~~ ~~через формулу  $(52^7 \cdot 10000) // 10000$~~

1) Округлим число  $52^7$  до точности 0,0001 при помощи ceil  $(52^7 \cdot 10000) / 10000$

2) Вычтем из этого числа 1, получив десятичную дробь, которую необходимо проверить с  $\frac{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

3) Так как наш путь проверено с точностью до 0,0001, но в какой-то моменте наш бюджет не вытеснит градус  $2 + \frac{1}{2}$  и мы можем округлить до 2,5 (не будет вытеснено, т.к. не будет вытеснено не округленное число)

4) Благодаря 3 пункту мы можем сказать, что теперь будем представлять градус таким образом, что будем добавлять к знаменателю градус, до тех пор пока округленное число не перестанет меняться. ~~Видно из предыдущего.~~ Это есть.   
 Первый наш шаг перевода будет  $\frac{2 + \frac{1}{2}}{2} = 0,4$   
 Следующий бюджет  $\frac{1}{2 + \frac{1}{2}} \approx 0,41667$  57

Число уменьшилось, значит добавим градус.

5) Когда наша градус не уменьша, то мы можем пока считать равенство градус, найденной в 4 пункте, и вычесть градус, найденной в 3 пункте.

13  
1) Пусть  $n$  — <sup>сторона</sup> ~~периметра~~  $a$  — наш шаг, которое задается, а  $b$  — шаг — число в 10 с. а

2) Переведем данное число из 16 с. в 10 с., ~~т.к. мы переводим в десятичную систему~~ <sup>способом, что переводим в десятичную систему</sup> ~~варианта~~  $a$ . Теперь  $b$  к заданному числу ~~сторона~~ <sup>сторона</sup> ~~группы~~  $\&_1$   $sizc()$ . Теперь введем массив.  $b[i]$   $k$ . Создаем цикл с условием  $(i=0, i < k, i++)$ . В нем цикл записываем.

```
if (a[i] != 1 && a[i] != 2 && a[i] != 3 && a[i] != 4 && a[i] != 5 && a[i] != 6 && a[i] != 7 && a[i] != 8 && a[i] != 9 && a[i] != 10) b[i] = a[i]
```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

if (b == A)
if (b == B)
if (b == C)
if (b == D)
if (b == E)
if (b == F)
++i;
c = c * b[i] * 10^i;

```

*анализ*  
*упр*

Поскольку мы знаем, что сумма цифр равна 10, то можно перебрать все возможные варианты. Для этого будем перебирать все это в сторону S:

```

for (i=0; i<=9; ++i)
{
for (j=0; j<=9; ++j)
{
S = S + "i";
for (k=0; k<=9; ++k)
{
S = S + "j";
}
S = S + "k";
}
}

```

4) Теперь мы переберем все возможные варианты в сумме. И оно задано в 2.  $count \ll S$ ;

1

- 1) Через цикл  $for(x=0; x \leq 9; ++x)$  мы будем перебирать  $x$
- 2) Через цикл  $for(y=0; y \leq 9; ++y)$  мы будем перебирать  $y$
- 3) Через  $for(z=0; z \leq 9; ++z)$  мы будем перебирать  $z$ .
- 4) Теперь, соединив все эти 3 цикла, мы сможем перебрать все возможные варианты, которые переберет все возможные  $x, y, z$ . Для того, чтобы проверить делится ли число на 10, мы можем просто найти остаток от деления, если он равен 0, то число делится.
- 5) Конкретный код:
 

```

for (x=0; x <= 9; ++x)
for (y=0; y <= 9; ++y)
for (z=0; z <= 9; ++z)
if ((20010002022004 + x * 10^9 + z * 10^8 + y * 10^7 + x * 10^6 + y * 10^5) % (y * 100 + x + 20) == 0)

```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

cout << x << " " << y << " " << z << endl  
Вывод программы будет ответом.

к 2

1) Пусть  $a = 64^{256}$ , переводим данное значение в строку s.

2) ~~Через функцию find("0") задаем в переменную b~~

3) ~~Создаем цикл с условием for(b=0; b < s.find("0"); b++)~~

3) В цикл цикла записываем ~~каждый~~

{ cout << (s.size() - b)

SEB } = "0" (мы заменили место s на 0)

}

4) ~~Теперь программа выведет преобразование s в число~~

~~$64^{256}$  в порядке возрастания~~

Олимпиада школьников «Надежда энергетики»

INFO2	Уч. ИСТАНЦИОННО, С ИСПОЛЗОВАНИЕМ ВКС
№ группы	Место проведения

GR 98-66
шифр

— Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73111

ФАМИЛИЯ МАТЕВОСОВА

ИМЯ АНАСТАСИЯ

ОТЧЕСТВО МИХАЙЛОВНА

Дата рождения 02.10.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 13 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: \_\_\_\_\_

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Задача №1

$$A \rightarrow C = 1$$

$$A \& B \rightarrow D = 1$$

$$\neg B \rightarrow E = 1$$

$$C = 0$$

$$D = 0$$

$$E = 0$$

1 - ИСТИННО

0 - ЛОЖНО

~~Сторона листа, не заполнять!~~

$$\bar{A} = \neg A$$

" + " ИЛИ

$$A \rightarrow C = \bar{A} + C$$

$$\begin{cases} A \rightarrow C = \bar{A} + C = 1 \\ C = 0 \end{cases}$$

" + " обозначает "ИЛИ",

$$\downarrow$$
  
$$\neg A + 0 = 1$$

$$\downarrow$$
  
$$\neg A = 1 \Rightarrow A = 0$$

$$\neg B \rightarrow E = \neg(\neg B) + E = B + E = 1$$

$$\begin{cases} B + E = 1 \\ E = 0 \end{cases} \Rightarrow B + 0 = 1 \Rightarrow B = 1$$

$$A \& B \rightarrow D$$

$$\begin{aligned} A \& B \rightarrow D &= \overline{A \& B} + D = \bar{A} + \bar{B} + D = 1 \\ \bar{A} + \bar{B} + D &= 1 \end{aligned}$$

По закону де Моргана

$$\overline{A \& B} = \bar{A} + \bar{B}$$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Продолжение №1

$$\begin{cases} \bar{A} + \bar{B} + D = 1 \\ A = 0 \\ B = 1 \\ D = 0 \end{cases}$$

$$\Rightarrow 1 + 0 + 0 = 1$$

верно



$$\begin{cases} A = 0 \\ B = 1 \end{cases}$$

Ответ:  $\begin{cases} A = 0 \\ B = 1 \end{cases}$

истина, логично?



ВНИМАНИЕ! Преворачается только то, что записано с этой стороны листа в рамке справа

Задача №2

алг Фробь(с)

НАЧ

вещ  $x, s, c, e, n, k$  $e = 1$  $x = 1/3$ 

повторять

ПОКА  $e \geq 0,00001$ 

НЦ

$$s = 1 / (2 + 1 / (1 + 1 / (3 + 1 / (1 + 1 / (2 + 1 / (3 + x))))))$$
 $x = s$ 

$$c = 1 / (2 + 1 / (1 + 1 / (3 + 1 / (1 + 1 / (2 + 1 / (3 + x))))))$$
 $x = c$  $e =$  ~~abs(s-c)~~  $abs(s-c)$ 

кц

 $n = \text{sqrt}(19)$  $k = 4 +$  ~~c~~  $c$ если  $abs(k-n) < 0,00001$  то~~вывод "Формула верна"~~

вывод "Формула верна"

иначе

вывод "Формула неверна"

все  
кон



ВНИМАНИЕ! Прозеряется только то, что записано с этой стороны листа в рамке справа

### Продолжение 2

Комментарии

Обозначим за  $K$  итог, который мы получили в правой части формулы

$K = 4 + c$ , где  $c$  — это бесконечная дробь

Заметим, что  $c = 1/2 + 1/(1+1/3) + 1/(1+1/2) + 1/(3+x)$

где  $x \rightarrow c$ , точнее при ка-л-в таких операциях, т.е. приодов вычисления  $c$ , стремящихся к бесконечности  $x \rightarrow c$ .

Чем больше раз мы будем продолжать эту операцию, тем точнее будет итоговый результат. У нас у дробей есть период

Запустим цикл, пока  $e \geq 0,000001$

(будем вычислять с чуть большей точностью)

Изначально возьмем  $x = 1/3$ , тогда

будем брать  $x = S$  — т.е. бесконечной

дроби, полученной на предыдущем

шаге. При увеличении числа вычислений, а

также т.к. ~~мы~~ мы будем чуть

большею точность  $0,000001$  приближаться

от величины  $x = 1/3$  станет еще меньше

(мы его изначально будем брать, т.к. с точностью

до десятых  $S$  равен этой величине). (увеличим

кач-ва вычислений приближение будем на каждом

$x$  таким будет уменьшаться. ~~мы~~ ~~будем~~ ~~вычислять~~  
 На каждой итерации цикла мы вычисляем



ВНИМАНИЕ! Проверьте только то, что записано с этой стороны листа в рамке справа

Продолжение №5

2 дробей  $s$  и  $c$ . А потом измерили  $e = \text{abs}(s-c)$  т.е. погрешность, насколько точнее стала вычисляться дробь.

После выхода из цикла вычисляем  $k=4+c$ , далее проверяем  $\text{abs}(k-n)$ , если величина разности по модулю  $< 0,0001$ , то формула верна, иначе она неверна.

$$n = \sqrt{19} = \text{sqrt}(19)$$

Выводим результат



ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа

Задача N4

```

алг Путь()
нач
цел s, i, j, N, k, z
цел table table[N, N]
s = 0
ввод N
если N <= 0 то
    вывод 'Неверное значение N'
иначе
    i = N
    j = N
    повторять
        пока i >= 1
        нц
            повторять
            j = N
            повторять
                пока j >= 1
                нц
                    ввод table [N, N]
                    j = j - 1
                кц
            i = i - 1
        кц
    i = 1
    повторять
        пока i <= N
        нц
            для j от 1 до N
            нц
                k = table[i, j]
                z = Proverka(k)
                s = s + z
            кц
        i = i + 2

```





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Продолжение №

~~ЕСЛИ  $i \bmod 2 = 0$  ТО  
 $k = \text{table}[i, 1]$   
 $t = \text{Proverka}(k)$   
 $S = S + t$   
 $k = \text{table}[i, N]$   
 $t = \text{Proverka}(k)$   
 $S = S + t$   
 ВСЕ~~

<sup>кц</sup>  
<sup>нач</sup> от 1 до N  
 ЕСЛИ  $i \bmod 4 = 0$  ТО  
 $k = \text{table}[i, 1]$   
 $t = \text{Proverka}(k)$   
 $S = S + t$

ВСЕ  
 ЕСЛИ  $i \bmod 4 = 2$  ТО  
 $k = \text{table}[i, N]$   
 $t = \text{Proverka}(k)$   
 $S = S + t$

ВСЕ  
~~кц~~  
 кц  
 ВЫБОР S  
 ВСЕ  
 КОН

алг Proverka (алг цел k)  
 НАЧ  
 цел a, f  
 ВЕЩ, r



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

f=0
a=2
r=sqrt(k)
Продолжение dy
если k mod 2=0 и k > 2 то
    f=1
ИНАЧЕ
    a=3
если k=1 то
    f=1
все
повторять
пока a < r+1
нц
если k mod a = 0 то
    f=f+1
    все
    кц
    a=a+2
все
если f=0 то
    вернуть k
ИНАЧЕ
    вернуть 0
все
кон

```

Комментарии:

Узнав, что существуют вояж массива, решил сделать так, чтобы нуль стоял или с правой нижней клетки



Проверки числа на простоту с помощью алгоритма Proverka.

Если  $k:d$ , то  $k:k$  т.е. парные делителями  $\rightarrow$  чтобы проверить, что число простое нужно проверить что оно не делится ни на одно число  $> 2$  и меньше квадратного корня из этого числа  $k$ .

Все строки с соседними номерами вояж будут прибавлять к сумме простое числа в них



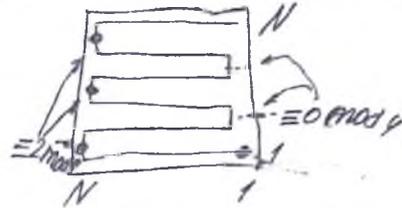
ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

### Продолжения

Также добавим клетки, которые мы не посчитали, но они выйдут в нуль и также проверим простоту

Это клетки в 1 и N столбце

В 1 столбце в номерах строк  $\equiv 2 \pmod 4$   
а в N столбце в номерах строк  $\equiv 0 \pmod 4$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

### Задача 15

Сначала выполним сортировку массива по возрастанию  $x$  <sup>1 строка</sup>  
 Функция монотонно возрастает, если при увеличении  $x_i$ ,  $y_i$  возрастает

Заведём 2 массива  $x_i$  и  $y_i$

$x_i$  - массив координат по  $x$  - 1 строка

$y_i$  - массив координат по  $y$  - 2 строка

Коор. по  $x$  - 1 строка таблицы  
Коор. по  $y$  - 2 строка таблицы

алг. функция  $f$   
нач

~~ввод~~ ~~вывод~~ вещ,  $q, f$

цел  $N, i, j, t, s, m$

вещ table ~~table~~  $[2, N]$

~~ввод~~  $N$

$i = 1$

для  $j$  от 1 до  $N$

нц

вывод table  $[i, j]$

кц

$j = 1$

для  $i$  от 1 до  $N$

нц

вывод table  $[i, j]$

кц





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

~~S=0~~ Продолжение  $s$

$i=1$

~~t=0~~ повторить

ПОКА  $t \leq N$  или  $S=0$

нц  ~~$s=0$~~   $j$  от 2 до  $N$

нц если  $table[1, j] \leq table[1, j-1]$  то

$f = table[1, j-1]$

$table[1, j-1] = table[1, j]$

$table[1, j] = f$

$g = table[2, j-1]$

$table[2, j-1] = table[2, j]$

$table[2, j] = g$

~~$S=S+1$~~

все

кц

кц

~~$m=0$~~  повторить

ПОКА  $j$  от 2 до  $N$

нц

если  $table[2, j-1] > table[2, j]$

или ( $table[2, j-1] = table[2, j]$  и

$table[1, j-1] < table[1, j]$ ) то

$m = m+1$

нц

кц





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Продолжение 15

если  $m=0$  то

вывод "Функция является монотонно  
возрастающей"

ИНАЧЕ

вывод "Функция не является монотонно  
возрастающей"

все  
кОН



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Из клетки есть 4 варианта движения



заданы

т.е. если клетка имеет номер  $i, j$

то возможные клетки:

- 1)  $i = i + 3 \quad j = j - 1$
- 2)  $i = i + 3 \quad j = j + 1$
- 3)  $i = i + 1 \quad j = j + 3$
- 4)  $i = i + 1 \quad j = j - 3$

Реш

Рассмотрим все варианты движения, ~~которые~~

т.к. клеток конечное число и из каждой 4 варианта движения

Запустим циклы

Если ни под одной из конечных клеток нет суммики, то робот проиграл, т.к. разобьется

Если робот зашел, не дойдя до верхней клетки, то тоже проиграл

Если робот проиграл, то он не может добраться до указанной верхней клетки

Иначе может



# Олимпиада школьников «Надежда энергетики»

И11F02	Дистанционно, с использованием ВКС
№ группы	Место проведения

GR 98-22
----------

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

Вариант № 73111

ФАМИЛИЯ НИКОЛАЕВ

ИМЯ АЛЕКСАНДР

ОТЧЕСТВО АЛЕКСЕЕВИЧ

Дата рождения 19.08.2003

Класс: 11

Предмет Информатика

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 09 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

51

Запишем возможные комбинации A, B, C, D, E

A	B	C	D	E	$A \rightarrow C$	$A \& B \rightarrow D$	$\neg B \rightarrow E$
1	0	0	0	0	0	1	0
0	1	0	0	0	1	1	1
1	1	0	0	0	1	0	1
0	0	0	0	0	1	1	0

1 - истина, 0 - ложь

Подходит только строка, в которой  $A = 0, \& B = 1$

Ответ:  $A = 0, B = 1$   
(ложь) (истина)



54

Заметим, что чтобы попасть такой змейкой из левого нижнего угла в правый верхний угол, н должно быть нечётным.

Будем обходить таблицу по строкам с шагом 2 (счётчик  $i$ ). Каждую строку будем обходить слева направо, если  $i$  кратно 4, и справа налево в ином случае. При этом будем проверять числа в ячейках на простоту. ~~Пусть  $f$  будет функцией, возвращающей 1, если  $n$  простое, и 0 в противном случае.~~

Номера (числа) на собраных мишках будем записывать в таблицу выводим в том порядке, в котором они были собраны.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

54 (продолжение)

АЛГ Шашки()

НАЧ

ЧЕЛ  $n, \text{table}[n, n]$

ВВОД  $n$

Если  $n \bmod 2 = 1$  то

Вывод "Неверные исходные данные"

иначе

для  $i$  от 1 до  $n$

нч для  $j$  от 1 до  $n$

нч ВВОД  $\text{table}[i, j]$

кч

кч

для  $i$  от  $n$  до 1 шаг -2

нч Если  ~~$i \bmod 4 = 0$~~   $(n-i) \bmod 4 = 0$  то

для  $j$  от 1 до  $n$  шаг 1

нч Если простое( $\text{table}[i, j]$ ) то

Вывод  $\text{table}[i, j]$

~~всё~~

кч

иначе

для  $j$  от  $n$  до 1 шаг -1

нч

Если простое( $\text{table}[i, j]$ ) то

Вывод  $\text{table}[i, j]$

кч ~~всё~~

Если  $i < 1$

Если  ~~$(n-i) \bmod 4 = 0$~~  ~~простое~~ то

Если простое( $\text{table}[i-1, n]$ ) то

Вывод  $\text{table}[i-1, n]$

всё

иначе

~~Вывод~~

Если простое( $\text{table}[i-1, 1]$ ) то

Вывод  $\text{table}[i-1, 1]$

всё

кон



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

54 (продолжение)

Диаг простое (арг цел  $n$ )

на  $z$

цел  $i$

если  $n \leq 1$  то

вернуть ложь

всё

если  $n = 2$  или  $n = 3$  или  $n = 5$  или  $n = 7$  то

вернуть истина

всё

если  $n \bmod 2 = 0$  то

вернуть ложь

всё

если  $n \bmod 3 = 0$  то

вернуть ложь

всё

для  $i$  от 5 до целая-часть( $n$ ) шаг 6

нз

если  $n \bmod i = 0$  то

вернуть ложь

всё

если  $n \bmod (i+2)$  то

вернуть ложь

всё

кз

вернуть истина

кон

(7)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

53

1) Четыре функции:



checkTopLeft(x, y, steps) - сильно вверх, путок влево  
 checkLeftTop - сильно влево, путок вверх  
 checkTopRight  
 checkRightTop

они принимают текущую позицию и набор доступных ступеней

и возвращают координаты целевой ступени (x; y), либо (-1; -1), если разбились

2) Реализация ~~каждой~~ такой функции:

алг checkTopLeft (arg yes x, arg yes y, arg yes k,  
 // вверх 3, влево 1  
 // k - длина массива ступеней  
 (arg yes steps[k, 2])

для i от y+1 до y+3  
 ну

если steps содержит (x; i) то

вернуть (-1; -1) // ударились головой о ступеньку - умирать

вii  
 кy

yes xNew = i - 1

yes yNew = x - 1 // после прыжка у нас эти координаты (для других функций будет x+1, x-3, x+3)

если xNew < 0 или xNew > xMax или yNew > yMax то вернуть (-1; -1) // выпали за границу и разбились

// xMax, yMax - размеры поля  
 всё



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

53 (продолжение)

для  $j$  от  $y_{New}$  до  $0$  шаг  $-1$

нч

если steps содержит  $(x_{New}, j)$  то

вернуть  $(x_{New}, j)$  // под нами после прыжка  
оказалась ступень -  
приземляемся на неё

всё

кч

g = 0

вернуть  $(-1, -1)$  // ступени не оказалось - разбились  
кон

(аналогично реализуем три другие функции)

3) Рекурсивная функция  $main(x, y, steps)$

принимает текущую позицию, возвращает  
bool / false - разбились, выиграли

(Пусть  $x_{Max}$  и  $y_{Max}$  будут глобальными переменными)

алг  $main(x, y, steps)$

нат

ули steps[3] =  $checkTopLeft(x, y, steps)$

ф. если  $step = fin$  то //  $fin$  - финальная правая  
вернуть истина верхняя ступень

всё

~~если~~

если  $step < > (-1, -1)$  то

лог  $res = main(steps[0], steps[1], ~~steps~~ steps - (x, y))$

// вызываем рекурсивно себя. из списка ступеней  
убираем  $step$ , на которой стоим, чтобы избежать  
зацикливания.

если  $res$  то

вернуть истина

всё

(3 раза аналогично для 3-х других функций)

вернуть ложь // ни один из 4 вариантов не сработал

кон



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

52

$$\text{число } 1 = 4 + \frac{1}{A_1} = x_1$$

$$\text{где } A_1 = 2 + \frac{1}{1 + \frac{1}{3 + \frac{1}{1 + \frac{1}{2 + \frac{1}{8}}}}}$$

разница 1 =  $|(x_1)^2 - 19|$  если она  $< 0,01$ , то формула верна, конец (0,01, а не 0,0001, т.к. сравниваем квадраты) иначе идем дальше.

$$\text{число } 2 = 4 + \frac{1}{A_2} = x_2$$

$$\text{где } A_2 = 2 + \frac{1}{1 + \frac{1}{3 + \frac{1}{1 + \frac{1}{2 + \frac{1}{A_1}}}}}$$

Разница 2 =  $|(x_2)^2 - 19|$  если она  $< 0,01$ , то формула верна, конец

$$\text{число } 3 = 4 + \frac{1}{A_3} = x_3$$

$$\text{где } A_3 = 2 + \frac{1}{1 + \frac{1}{3 + \frac{1}{1 + \frac{1}{2 + \frac{1}{A_2}}}}}$$

Разница 3 =  $|(x_3)^2 - 19|$  если она  $< 0,01$ , то формула верна, конец

И так. далее. Если в какой-то моменте разница (и)  $>$  разница (и-1), то формула неверна, конец



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

55

- Возьмём алгоритм быстрой сортировки точек по координате  $x$ .
- в нём при каждом сравнении  $x_1$  и  $x_2$  проверяем соответствующие  $y_1$  и  $y_2$
- Если в какой-то момент видим, что  $x_1 > x_2$ , а  $y_1 \leq y_2$ , то прерываем выполнение и возвращаем ложь. Также возвращаем ложь, если  $x_1 = x_2$  и  $y_1 \neq y_2$  (ложь - ф-я не возрастает)
- Если дошли до конца, то функция возрастает.





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Алг  
нат default Sorting Algorithm (a, b) // сравнение двух значений a[0] - x и a[1] - y

если a[0] < b[0] то

если a[1] > b[0] то

~~вернуть~~  
вернуть -99 // не возращает  
всё  
всё

если a[0] > b[0] то

если a[1] <= b[0] то

вернуть -99  
всё  
всё вернуть 1  
вернуть 0

кон

Алг  
нат quick Sort (unsorted Array)

sorted Array = копия (unsorted Array)

Функция recursive Sort (start, end)

нат

если (end - start) < 2 то

вернуть

всё

pivotVal = sorted Array [end]

splitIndex = start

while i от start до end-1

sort = default Sorting Algorithm (sorted Array [i], pivotVal)

if если sort = -99 то  
вернуть -99

если sort = -1

если splitIndex < i

temp = Sorted Array [splitIndex]



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

55

sorted Array [split+index] = sorted Array [i]  
 sorted Array [i] = temp  
 всё

split+index += 1

~~кон~~  
 всё

~~кон~~  
 всё

sorted Array [end] = sorted Array [split+index]  
 sorted Array [split+index] = pivotVal

if recursive Sort (start, split+index - 1) = -99 но  
 вернуть -99

всё

если recursive Sort (split+index + 1, end) = -99 но  
 вернуть -99

всё

кон

если recursive Sort (0, unsorted Array.length - 1) = -99 но  
 вернуть ничего // не возвращаем

всё

вернуть sorted Array // возвращаем  
 кон

# Олимпиада школьников «Надежда энергетики»

I11F01	Авистанционно с ПРИМЕНЕНИЕМ ВКС
--------	------------------------------------

№ группы

Место проведения

GR 12-64
----------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73111

ФАМИЛИЯ ПАСКОНОВА

ИМЯ ЛЮАМИЛА

ОТЧЕСТВО ВЛАДИМИРОВНА

Дата рождения 31.07.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 11 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:

Л.Паск.

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N1

$$A \rightarrow C = 1$$

$$A \& B \rightarrow D = 1$$

$$\bar{B} \rightarrow E = 1 \quad (\text{по условию}) \quad \text{Найти: } A, B \text{ (?)}$$

$$C = 0$$

$$D = 0$$

$$E = 0$$

$$A \rightarrow C \Leftrightarrow \bar{A} \vee C$$

$$A \& B \rightarrow D \Leftrightarrow \overline{A \& B} \vee D \Leftrightarrow \bar{A} \vee \bar{B} \vee D$$

$$\bar{B} \rightarrow E \Leftrightarrow B \vee E$$

$$\left\{ \begin{array}{l} \bar{A} \vee C = 1, C = 0 \Rightarrow A = 0 \\ \bar{A} \vee \bar{B} \vee D = 1 \end{array} \right.$$

$$\left\{ \begin{array}{l} B \vee E = 1, E = 0 \Rightarrow B = 1 \end{array} \right.$$

A - ложно, B - истинно

Ответ: A - ложно, B - истинно

(+)

Примечание к задачам 2-5

// - это комментари, не относятся к решению



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N2

Алгоритм: Рассмотрим числа  $4, 4 + \frac{1}{2}; 4 + \frac{1}{2 + \frac{1}{2}};$   
 $4 + \frac{1}{2 + \frac{1}{1 + \frac{1}{3}}}; 4 + \frac{1}{2 + \frac{1}{1 + \frac{1}{3 + \frac{1}{4}}}}$  ..... пока в какой-то момент  
 это число не попадет в интервал  $[\sqrt{19} - 0,0001; \sqrt{19} + 0,0001]$ .

Алг Кривыначвещ  $n = \text{sqrt}(19)$ цел  $a[6]$  $a[0] = 2$  $a[1] = 1$  $a[2] = 3$  $a[3] = 1$  $a[4] = 2$  $a[5] = 8$ вещ  $x = 4$ вещ  $\rho = 2$ ; цел  $k = 1$  // индикатор итерацийпока ( $x \leq n - 0,0001$  или  $x \geq n + 0,0001$ ) $x = x + \frac{1}{\rho}$  $\rho = \rho + \frac{1}{a[k]}$  $k = (k + 1) \bmod 6$ к.цвывод "Формула справедлива"



Олимпиада школьников «Надежда энергетики»

Вариант: 73111

ШИФР, НЕ ЗАПОЛНЯТЬ! ⇨

GR12-64



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

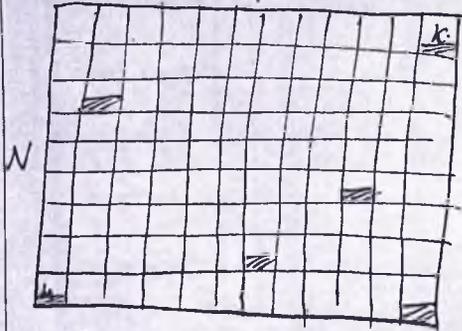
кон

Программа выведет необходимый результат, иначе закончится из-за превышения тайм-лимита (тогда эта горюшка неверна)

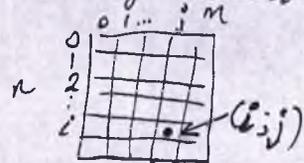


ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3 Вертикальная стена размером  $M \times N$   
( $N$  «строк»,  $M$  «столбцов»)



→ клетка со ступенькой



строки столбцы

Пусть стена — это матрица  $a[N+2; M+2]$   
0 — ступеньки нет, 1 — ступенька есть в каждой её  
ячейке, индексы нумеруются с 0  
на языке с помощью кавычек.

АЛГ Матрица

нач

~~Для (цел i=0; i<n; ++i) цел n, m~~

~~Для (цел j=0; j<m; ++j)~~

Ввод  $n, m$ . то цел  $a[n][m]$  // массив

Если  $n \leq 0$ , то вывод «ошибка ввода» всё, конец прочт.

Если  $m \leq 0$ , то вывод «ошибка ввода» всё, конец прочт.

Для (цел  $i=0; i<n; ++i$ )

ку

Для (цел  $j=0; j<m; ++j$ )

ку

Ввод  $a[i][j]$ . Если  $(a[i][j] \neq 0 \text{ и } a[i][j] \neq 1)$ , то

ку

ку

вывод

«ошибка  
ввода»  
всё, конец прочт

Если  $a[n-1][0] = 0$ , то вывод «ошибка ввода», конец прочт.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Если  $a[0][m-1]=0$ , то вывод "ошибка ввода",  
конец прогн все

~~и~~ // если на  $n$  и  $k$  нет ступенек продолжит дальше

Если робот сейчас находится в клетке  $[i][j]$ , то он может переместиться либо в клетку  $[i-1][j+3]$ , либо в клетку  $[i-3][j+1]$ , либо в кл.  $[i-1][j-3]$ , либо в кл.  $[i-3][j-1]$ . Сначала он находится в клетке  $a[n-1][0]$ .

Будем рассматривать всевозможные шаги робота, для каждого проверяя, не шагает ли он на этом этапе.

Функция АЛГ проверка (АРГ цел  $i$ ; цел  $j$ ) нет ступенек на вертикали  
нач если  $(i=0$  и  $j=m-1)$ , то вывод "Может" конец программы (окак)  
если  $(i-1 \geq 0$  и  $(j+3) < m$  и  $a[i-1][j]=0$  и хотя бы одно из чисел

для цел  $i_x$  от  $(i-1)$  до  $(n-1)$   $a[i_x][j+3]=1$

// т.е. есть ступенька на  $[i-1][j+3]$ , или  $[i][j+3]$ , или...

...  $[n-1][j+3]$ , т.е. робот не шагает

ТО проверка  $(i-1; j+3)$  все

~~иначе~~ если  $(i-3) \geq 0$  и  $(j+1) < m$  и  $a[i-3][j]=0$  и

$a[i-1][j]=0$  и  $a[i-2][j]=0$  ~~и~~ и для цел  $i_x$  от  $(i-3)$  до  $(n-1)$  хотя бы одно из чисел  $a[i_x][j+1]=1$ ,

ТО проверка  $(i-3; j+1)$  все

если  $(i-1) \geq 0$  и  $(j-3) \geq 0$  ~~и~~ и  $a[i-1][j]=0$  и для цел  $i_x$  от  $(i-1)$  до  $(n-1)$  хотя бы одно из чисел  $a[i_x][j-3]=1$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

то проверка  $(i-3; j-3)$ 

если  $(i-3) \geq 0$  и  $j-3 \geq 0$  и  $a[i-3][j] = 0$  и  $a[i-1][j] = 0$  и  $a[i-2][j] = 0$  и для всех цел.  $i_x$  от  $(i-3)$  до  $(i-1)$  хотя бы одно из  $a[i_x; j-1] = 1$ ,  
то проверка  $(i-3; j-1)$

кон

Основной код (продолжение)\*

Проверка  $(n-1; 0)$ ;

И если возможно робот дойдёт, до  $60$ -ции  
это появится как надпись на стене. кричу  
вывод "не может" // в  $90$ -ции перехода всех  
случаев робот не мож. добр. до  $k \Rightarrow$  напишем  
ответ от руки

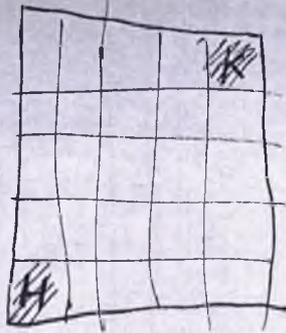
кон

Таким образом, алгоритм следующий: рассматриваем всевозможные траектории робота и если в какой-то момент на одной из таких траекторий робот ломается (выход за границу пола / касание стены сверху вертикали / касание стены снизу до конца стены), то эта траектория вычеркивается. Если в ходе какой-то траектории робот приходит на финишную клетку, то робот дошёл. Если ни в одной траектории робот не добр. до конечной клетки (ломается), то робот не может дойти до клетки  $k$ .



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N4



Алгоритм: проходим по всем клеткам проходов, по которым можно идти (в матрице  $a[m][n]$  такие клетки обозначены 1) и собираем те самые «прямые шимки». Исходя из условия, у нас только один путь прохождения лабиринта.

Проверка на простоту

АЛГ simple (цел  $n$ )

нач

для (<sup>цел</sup>  $i=2; i \cdot i \leq n; ++i$ ) // проверка до корня из  $n$

к.у.

если ( $n \bmod i = 0$ ), то //  $n$ -составное

конец программы, без вывода (break)

к.у.

возврати ( $n$ ) // ф.ция return,  $n$ -простое.

конец

АЛГ шимки

нач ~~цел~~ ~~н~~ <sup>цел</sup>  $N; \text{ввод } N; \text{цел } \text{в } [m; n], \text{цел } x$

для (цел  $i=0; i < n; ++i$ ) ~~к.у.~~

к.у. для (цел  $j=0; j < n; ++j$ )

к.у.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

ВВОД  $x \in \{1, 0\}$ , можно/нельзя идти?

~~Если  $x=1$  то~~

ВВОД  $b[i; j]$  // значение на шине



Если  $x=1$  то вывод `simple` ( $b[i; j]$ )

// если  $b[i; j]$  - простое, то оно и выведется -  
// ответ на вопрос про простые или нет,  
// иначе вывода не будет.

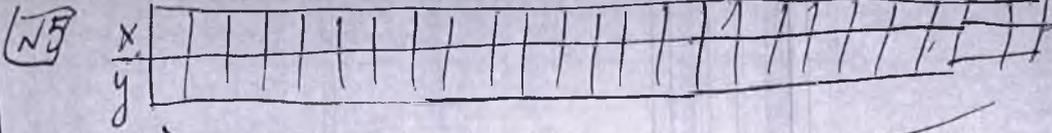
к.у

к.у

к.о.н



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



Алгоритм проверки: отсортируем массив абсцисс и массив ординат и проверим, соотв. ли массивы старых индексов ~~каждого~~ каждого эл-та массива

$x$  старому индексу каждого эл-та массива  
 $y$  (массив старых индексов — индексы от 0 до  $n-1$ , отсортированные в порядке сортировки массивов). Элемент ~~каждого~~

Быстрая сортировка: массив ~~тогда~~ массив индексов

Алг  $qsort$  (~~цел~~ <sup>вещ</sup>  $a[]$ , цел  $b[]$ , цел  $size$ ) ← <sup>размер массива</sup>

нач

цел  $i=0$

цел  $j=size-1$

цел  $mid = a[\frac{size}{2}]$

~~Алгоритм~~ повторяй

пока ( ~~$a[i] < mid$~~   $a[i] < mid$ )

кц

$i++$   $\{\{i=i+1\}$

кц

пока ( $a[j] > mid$ )

кц

$j--$   $\{\{j=j-1\}$

кц

Если ( ~~$i < j$~~   $i < j$ ) то



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

вещь tmp = a[i] ; вещь tmp = b[i]

a[i] = a[j]      b[i] = b[j]

a[j] = tmp      b[j] = tmp

i++

j--

всё

пока (i < j)

Если (j > 0) то

$\text{sort}(a[j], b[j], j+1)$

всё

Если (i < size) то

$\text{sort}(a[i], b[i], size-1)$

всё

конец

// рекурсивные вызовы,  
еще осталось, что  
сортировать

АЛГ Монотонность

НАЧ

~~вещь~~ вещь цел n ; ~~вещь~~

ввод n ; вещь x[n+2], y[n+2] ; цел vx[n+2], vy[n+2]

для (цел i=0; i < n; i++)

кч ввод x[i]

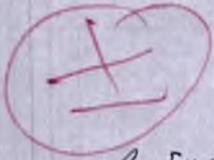
кч ввод y[i]

// ввод массива адресов и аргументов

~~для~~ для (цел i=0; i < 0; i++)

кч vx[i] = i

кч vy[j] = i // ввод массивов индексов





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

$qsort(x, vx, n)$

$qsort(y, vy, n)$

~~for~~ while (while  $i=0; i<n; i++$ )

кx

если  $vx[i] \neq vy[i]$ , то вывод «Функция немонотонна»  
конец программы (break)

кy

вывод «Функция монотонно возрастает»

### Олимпиада школьников «Надежда энергетики»

79F02 ДИСТАНЦИОННО,  
С ИСПОЛЬЗОВАНИЕМ ВЕБ

№ группы

Место проведения

VO 36-20

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73991

ФАМИЛИЯ Розачков

ИМЯ АНТОН

ОТЧЕСТВО Олегович

Дата рождения 19.05.2005

Класс: 9

Предмет Информатика

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 8 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№1.

$x, y, z$  - числа от 0 до 10, (или  $\sqrt{2x}$  должно быть 3 знаменными по  $y$  от 1 до 10)

переберём  $x, y, z$  (все  $f$  от (перебор) вкл. частн и леве и праве значение)

for  $x$  от 0...9

for  $y$  от 0...9 (или от ~~0~~ 1...9 если  $\sqrt{2x}$  должно быть 3 знаменными).

for  $z$  от 0...9:

$$A := 2 \cdot 10^{13} + x \cdot 10^{11} + 1 \cdot 10^{10} + z \cdot 10^9 + y \cdot 10^8 + x \cdot 10^7 + 2 \cdot 10^6 + 2 \cdot 10^4 + 2 \cdot 10^3 + x \cdot 10^2 + y \cdot 10 + 9$$

не факториал

$$B := y \cdot 10^2 + 2 \cdot 10 + x$$

if  $(A \% B == a)$  ("остаток от деления")



Запоминаем всевозможные

тройки  $x, y, z$ , можем сделать до массив если мы хотим их все запомнить

Так мы найдём все тройки (алгоритм займёт время порядка ~~10^3~~  $10^3$ ).



СХИМАТИЧЕСКИЕ РЕШЕНИЯ ПОДЛЕЖАТ  
ОЦЕНКЕ ТОЛЬКО В СЛУЧАЕ НЕОБХОДИМОСТИ

№  
~~нужно вывести строку~~  
~~сформированную из~~  
 string s = "46" (string - массив символов или  
 (массив байтов) - значение переменной s - строка  
 (или массив)

```
for i on 1 to 256:
  s = s + s
```

```
string s2 = s + s (повтор  

  строки s)
for i on 0 to s2.size() - 1:
  s2[i] = s2[i] * 4
```

```
s2[i] = (значение s2[i] * 4) % 10 + 9
g = [(значение s2[i] * 4) % 10]
" [g:1:4]" - значение генерации
```

```
for i on 1 to s2.size() - 1:
```



ВНИМАНИЕ! Проверяется только то, что написано с этой стороны листа в рамке справа.

$$s_{60}[i] = ((\text{цифра от } s_{60}[i]) \cdot 6) \% 10 + g$$

(цифра от  $s_{60}[i]$  - элемент стоящий в строке на месте  $i$  места помножится на 6)

$$g = \lfloor ((\text{цифра от } s_{60}[i]) \cdot 6) : 10 \rfloor$$

✓ (1) ~~формула~~: ~~н.к.~~ ~~регулярность~~

числа могут увеличиться по сравнению с первоначальными если  $g = 0$  то ~~тогда~~ ~~используем~~ увеличимся..

if  $g \neq 0$ :

$s_{60}$ .push\_back(g) (и push-back добавит в конец).

ну и обновим  $g$  перед следующим циклом

$g = 0$

~~это уже не формула!~~

if  $g \neq 0$ :

$s_{60}$ .push\_back(g)

$g = 0$

for  $i$  от  $0$  до  $s_{60}.size - 1$ :



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

if.  $s.size() - 1 \geq i$ :

$$s \leftarrow [i] = [(s_y + s_{60} + g) \% 10]$$

$$g = (s_y[i] + s_{60}[i] + g) \% 10$$

else:

$$s.push_back([(s_y + s_{60} + g) \% 10])$$

$$g = (s_y[i] + s_{60}[i] + g) \% 10$$

нам не нужно деление при проверке суммируемых  
или  $s_y[i]$  ~~не нужно делить~~  $s_{60}[i]$  ~~не нужно делить~~  
проверку!

~~то~~ и надо к концу строки добавить нули, мы  
имеем строку  $s$  ~~элементов~~ ~~массива~~ ~~массива~~  
 $s_y$   $s_{60}$ .

теперь мы можем проверить её элемент  
и если элемент равен 8 то вывести его к концу  
~~то~~ мы ~~записываем~~  $(i+1)$  ~~это~~ ~~будет~~  
позиция н.к. ~~нам~~ ~~мы~~ ~~знаем~~ ~~слева~~  
направо ~~но~~ ~~и~~ ~~то~~ ~~позиция~~ ~~будет~~ ~~идти~~  
по возрастанию  
(это задаём время порядка)

$256 \cdot 256$  ~~тогда~~ и ~~мы~~ ~~учитываем~~  
на не маленькую константу по ~~всему~~ ~~массиву~~  
10. ~~можно~~ ~~составить~~ ~~массив~~ ~~массива~~.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Главного десятиричного или же секунды.

N 3

Ну тут нам нужно просто для начала перевести наше 16-ричное число в десятичное а потом в десятичную С.С.

число a - это ~~то же самое~~ наше число в 16 ~~р~~ ричной системе счисления (иногда удобнее будет если оно будет предметивно в виде строки но даже если пожелаете использовать алгоритм предположим что это будет шестеричное число его ~~тоже~~ можно перевести в строку.

string s;

~~a = 0;~~

~~for (int i = s.size() - 1; i >= 0; i--)~~

~~sh = 1;~~

sh = 1;

for (int i = s.size() - 1; i >= 0; i--)

a = a + ~~(int) (s[i] - '0')~~ \* sh

sh = sh \* 16;

~~return a;~~







с этой стороны листа в рамке справа

$$a = 2.$$

формула того же вида:

$$a = (1:a) + 2$$

$$a = 1 + (1:d).$$

таким образом мы получим значение  $\sqrt{2}$  мы  
10 раз через. для получения  $\sqrt{2}$  мы  
можем просто запросить его у компьютера  
через специальную функцию или  
нашем языке программирования

в C++ это можно сделать через

$$\text{sqrt}(2)$$

в python это можно сделать через

$$2**(1/2)$$

~~в python~~

$$v = \text{sqrt}(2)$$

$$v = [v : 0, 0001]$$

$$a = [a : 0, 0001]$$

~~проверяем~~

$$if v == a:$$

то формула работает

иначе не работает.



} нужно чтобы  
оставить точность  
0,0001.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

н.с.

~~A~~  $x \rightarrow y$  я понимаю как ~~значимый~~  
 $x$  передается в  $y$ . ?

Если с логичное по т.к. ~~с~~  $C$  было  
совершенна ~~и~~ только  $A \rightarrow C$  то значим  
 $A$  - логично.

т.к. если ~~A~~  $E$  логичное по т.к. с ~~с~~  $E$  было  
совершенна только  $\neg B \rightarrow E$  то  
 $B$  - истина.

тогда если  $A$  - логично а  $B$  - истина то все  
условия выполняются.

(±)

# Олимпиада школьников «Надежда энергетики»

Г11F01	ПУСТАНЦИОННО, С ИСПОЛЬЗОВАНИЕМ ВКС
--------	---------------------------------------

№ группы

Место проведения

GR12-88
---------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73111

ФАМИЛИЯ Рощупкин

ИМЯ Андрей

ОТЧЕСТВО Владимирович

Дата рождения 05.09.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:

AB

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N1

Термоформулы утверждения в матрице (0,1) строят:

$$\begin{cases} A \leq C \\ A \cdot B \leq D \\ \neg B \leq E \end{cases} \Leftrightarrow \begin{cases} E=0 \\ D=0 \\ C=0 \end{cases} \begin{cases} A=0 \\ A \cdot B=0 \\ \neg B=0 \end{cases} \Leftrightarrow \begin{cases} A=0 \\ B=1 \end{cases}$$



Ответ: A - лгать; B - истина

N2

н.к.ч., дробь

~~res1~~ res1, res2 // два сравниваемых числа формата double

res1 = корень(19) - 4 // 19 с возмещением четверки из дроби

res2 = Prod(0) // первая итерация дроби

ПОКА (res2 + 0.0001 >= res1 и res2 - 0.0001 < res1) и i < 1000

res2 = Prod(res2) // рекурсивно углубляется в дробь

i++ // счетчик для прерывания цикла

к.ц.

если i >= 1000

вывод("дробь неверна")



иначе

вывод("дробь верна")

к.о.н

с.л. Prod (arg ~~arg~~ ch)

.НАЧ

дробь result // поделит дроби

result = 1 / (2 + 1 / (1 + 1 / (3 + 1 / (1 + 1 / (2 + 1 / (8 + ch))))))

вернуть (result)

к.о.н



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3

нат

```

цел mas M, N
вывод (M, N)
цел pole[M][N]
цикл i от 0 до M-1
  цикл j от 0 до N-1
    вывод (pole[i][j])
если search (pole, 0, N, [])
  вывод ("может")
иначе
  вывод ("не может")

```



ком

```

цел search (arr конет mas, arr цел x, arr цел y, arr way)

```

нат

```

если y < 0 или x < 0 или x ≥ mas.длина() или y ≥ mas[0].длина()
  вернуть ложь // кейс-выход за границы
если mas[x][y] != 1
  вернуть search (mas, x, y+1, way + [(x, y)]) // робот может влезть
иначе
  если y = 0 y = 0 и x == mas.длина()
    вернуть истина // попал в К
  res = ложь; new цел newway [4] [2] = [(x+3, y-1), (x+1, y-3), (x-3, y-1),
  если [(x+3, y+1)] не в way // (x-1, y-3)]
    res = res или search (mas, x+3, y+1, way + newway)
  если [(x+1, y+3)] не в way
    res = res или search (mas, x+1, y+3, way + newway)
  если [(x-3, y-1)] не в way
    res = res или search (mas, x-3, y-1, way + newway)
  если [(x-1, y-3)] не в way
    res = res или search (mas, x-1, y-3, way + newway)
  вернуть res

```

ком

// Можно оптимизировать алгоритм проверки на выход за границы до затычка границы, но об этом пока не задумывался



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

НЧ  
кол

```

цел N, кол
вывод(N)
цел mas[N][N], ans[N*N]
цел i от 0 до N-1
цел j от 0 до N-1
    вывод(mas[i][j])

```

```

цел i от N-1 до (N+N%2)/2 + 1
    если прост(mas[i][0]) // проходим от N до 1 (невозможны)
        кол += 1
        ans[кол-1] = mas[i][0]

```

```

цел j от 0 до N-1 // проходим от 1 до 2
    если прост(mas[(N+N%2)/2][j])
        кол += 1

```

```

        ans[кол-1] = mas[(N+N%2)/2][j]
цел i от 0 до (N+N%2)/2 - 1 // от K до 2 (невозможны)
    если прост(mas[i][N-1])
        кол += 1
        ans[кол-1] = mas[i][N-1]

```

```

вывод("простых чисел: " + кол) // кол - это количество чисел с простыми числами
вывод(ans[0:кол]) // выводим все эти числа

```

```

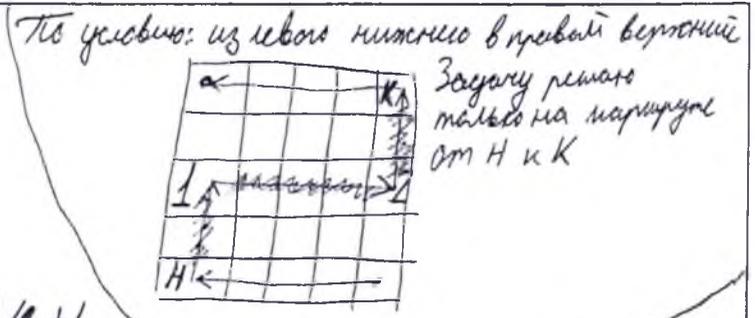
кон
алл прост(цел число)
кон

```

```

алл прост(цел число)
цел i от 2 до цел(корень(число)+1) // оптимальная проверка на простоту до sqrt, работает за O(sqrt(N))
    если число % i == 0
        вернуть ложь
    вернуть истину
кон

```



(7)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N5

ком

цел N

ввод(N)

цел mas[N][2]

цел i от 0 до N-1

ввод(mas[i][0], mas[i][1])

mas = sort(mas, key=mas[0]) // для дальнейшей проверки

цел flag = ложь

цел i от 0 до N-2

если mas[i][1] &gt; mas[i+1][1]

flag = истина

если flag

вывод("функция не монотонно возрастает")

иначе

вывод("функция монотонно возрастает")

ком

// Поскольку N по условию может быть очень большим сортировка массива через быструю сортировку, а при обходе массива выходим из цикла если хотя бы одна пара точек не соответствует правилу монотонности.

\* - используем > т.к. монотонность может быть нестрогой.

// f(x) является монотонно возрастающей если для любых  $x_1 < x_2 \in X$  имеем выполнение  $f(x_2) \geq f(x_1)$

// Предположим, что точки в массиве не отсортированы. Для удобства обозначим  $x_n$  и  $y_n$  в порядке

// отсортируем массив по значениям x для дальнейшей проверки

// Проверим  $f(x_2) \geq f(x_1)$   
// иначе выходим из цикла  
// запомнил flag

(F)

# Олимпиада школьников «Надежда энергетики»

I10F01	Дистанционно, с использованием ВКС
№ группы	Место проведения

QV 97-34
----------

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

Вариант № 73101

ФАМИЛИЯ САНДРОВ

ИМЯ КИРИЛЛ

ОТЧЕСТВО АЛЕКСЕЕВИЧ

Дата рождения 03.04.2004

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 9 листах

Дата выполнения работы: 28.02.21  
(число, месяц, год)

Подпись участника олимпиады:

Сандр

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N 1

При данных условиях логические выражения можно записать так:

$$\begin{cases} A \rightarrow C = 1 \\ A \& B \rightarrow D = 1 \\ \bar{B} \rightarrow E = 1 \end{cases}$$

Чтобы найти значения  $A$  и  $B$ , нужно рассмотреть при каких условиях выполняются выражения в данной системе:

1)  $A \rightarrow C = 1$

Т.к.  $C = 0$  по усл.:

$$A \rightarrow 0 = 1$$

Т.к. выражение импликации при  $1 \rightarrow 0 = 0$ , то:

$$A \neq 1$$

значит,  $A = 0$

2)  $A \& B \rightarrow D = 1$

Аналогично предыдущему выражению; т.к.  $D = 0$ :

$$A \& B \neq 1$$

Это всегда так, т.к. из прошлого выражения  $A = 0$ , и, если при операции конъюнкции один элемент равен нулю, то всё выражение ложно.

3)  $\bar{B} \rightarrow E = 1$

При  $E = 0$  (по усл.):

$$\bar{B} \rightarrow 0 = 1$$

При  $B = 0$  выражение  $\bar{B}$  будет равно 1, и  $1 \rightarrow 0 = 0$ , что не выполняет равенство.

значит,  $B = 1$

ОТВЕТ:  $A$  равно 0,  $B$  равно 1.

⊕  
логич, истина?



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N2.

Так как выражение  $\left(2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{4 + \dots}}}}\right)$  ТРЕБУЕТ ПОСЛЕДОВАТЕЛЬНОЕ ВЫЧИСЛЕНИЕ ЗНАМЕНАТЕЛЯ ВТОРОГО СЛАГАЕМОГО, ТО МОЖНО РЕАЛИЗОВАТЬ РЕКУРСИВНУЮ ФУНКЦИЮ, КАЖДЫЙ ВЫЗОВ КОТОРОЙ БУДЕТ НАХОДИТЬ ЗНАЧЕНИЕ ТЕКУЩЕГО ЗНАМЕНАТЕЛЯ ДРОБИ. РАЗ ЧИСЛО 4 В ЗНАМЕНАТЕЛЕ ПОВТОРЯЕТСЯ НА КАЖДОМ ЧЕТВЕРТОМ ВЫЧИСЛЕНИИ, ТО МОЖНО ПЕРЕДАТЬ В ФУНКЦИЮ ПЕРЕМЕННУЮ-СЧЕТЧИК ИТЕР, КОТОРАЯ БУДЕТ ПОКАЗЫВАТЬ, СКОЛЬКО РАЗ БЫЛА РЕКУРСИВНО ВЫЗВАНА ФУНКЦИЯ. Если ИТЕР = 4, а также, если ИТЕР делится на 4, то, значит, слагаемым будет четверка, если же ИТЕР = 0, то мы на первом шагу, и первое слагаемое равно двум, в остальных случаях слагаемое равно единице

алг Проверка

нач

вещ ответ, нужно, точность

ответ := решение(0)

нужно := sqrt(7)

точность := 0.0001

если abs(ответ - нужно) &lt;= точность

то вывод "Формула верна"

иначе вывод "Формула ошибочна"

все

кон

алг вещ решение(цел итер) | рекурсивная функция

нач

если итер &gt; 100

то знач := 1 / 1 | 101 не делится на 4

иначе если итер = 0

то знач := 2 + (1 / решение(итер + 1))

иначе

если mod(итер, 4) = 0

то знач := 4 + (1 / решение(итер + 1))

иначе знач := 1 + (1 / решение(итер + 1))

все

все

все

кон



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N2 (ПРОДОЛЖЕНИЕ)

Данная рекурсивная функция ограничена 100-а выполнениями, чтобы не быть бесконечной. При желании это ограничение можно увеличить для повышения точности. В приведённом алгоритме использованы следующие встроенные функции:

$\text{sqrt}(\text{цел число})$  - квадратный корень целого числа

$\text{mod}(\text{цел } a, \text{цел } b)$  - остаток от деления целых  $a$  и  $b$

$\text{abs}(\text{цел вещь } a)$  - вернуть результат по модулю  $|a|$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3.

Напишем функцию, имитирующую передвижение робота по клеткам. Проверяем, не выходит ли робот за границы поля или на колонну И, Если ещё робот не бывал в данной клетке поля, то вызываем эту функцию рекурсивно с обновлённым положением робота. Если робот сумел перейти из начальной плиты в конечную, то он смог пройти через препятствия, и ответ - ДА, иначе - НЕТ.

алг Робот

нач

цел  $m, n, i, j, nx, ny$ 

лог ответ

сим таб  $a[m+1, n+1]$ лог таб  $was[m+1, n+1]$ ввод  $m, n$ если  $m \leq 0$  или  $n \leq 0$ 

то вывод "НЕВЕРНЫЕ РАЗМЕРЫ ПОЛЯ"

иначе

нц для  $i$  от 1 до  $m$  шаг 1 | вводим клетки полянц для  $j$  от 1 до  $n$  шаг 1ввод  $a[i, j]$  $was[i, j] := \text{НЕТ}$ если  $a[i, j] = \text{'И'}$  то $nx := i$  | запоминаем начальное положение $ny := j$ 

все

кц ответ := решение( $nx, ny, m, n, a, was$ )

если ответ = ДА

то вывод "РОБОТ СМОЖЕТ ДОБРАТЬСЯ"

иначе вывод "РОБОТ НЕ СМОЖЕТ ДОБРАТЬСЯ"

все

все

кон

алг лог решение(цел  $x, y, \text{цел } m, \text{цел } n, \text{симтаб } a, \text{логтаб } was$ )

нач

если  $a[x, y, x] = \text{'К'}$ 

то знач := ДА

иначе

...

⊕



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3 (продолжение)

если  $y+1 \leq m$  и  $a[y+1, x] \neq 'X'$  и  $was[y+1, x] = \text{НЕТ}$

то ~~знач~~  $was[y+1, x] := \text{ДА}$

ВСЕ  $\text{ЗНАЧ} := \text{РЕШЕНИЕ}(x, y+1, m, n, a, was)$

если  $y-1 \geq 1$  и  $a[y-1, x] \neq 'X'$  и  $was[y-1, x] = \text{НЕТ}$

то  $was[y-1, x] := \text{ДА}$

ВСЕ ~~PE~~  $\text{ЗНАЧ} := \text{РЕШЕНИЕ}(x, y-1, m, n, a, was)$

если  $x+1 \leq n$  и  $a[y, x+1] \neq 'X'$  и  $was[y, x+1] = \text{НЕТ}$

то  $was[y, x+1] := \text{ДА}$

ВСЕ  $\text{ЗНАЧ} := \text{РЕШЕНИЕ}(x+1, y, m, n, a, was)$

если  $x-1 \geq 1$  и  $a[y, x-1] \neq 'X'$  и  $was[y, x-1] = \text{НЕТ}$

то  $was[y, x-1] := \text{ДА}$

ВСЕ  $\text{ЗНАЧ} := \text{РЕШЕНИЕ}(x-1, y, m, n, a, was)$

ВСЕ

$\text{ЗНАЧ} := \text{НЕТ}$

ВСЕ

КОН



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

НЧ.

Для начала поймём, как нужно двигаться по лабиринту, чтобы достичь ЛЕВОГО ВЕРХНЕГО УГЛА из ПРАВОГО НИЖНЕГО. Если  $N$  нечётно, то нужно, повернув, пройти 3 клетки по горизонтали, а затем продолжить по вертикали до следующей стены. Если  $N$  чётно, то нужно пройти 2 клетки по ~~вер~~горизонтали. Далее на каждой клетке нужно проверить, простое ли в ней число. Для этого реализуем функцию, проверяющую на простоту число. В ней переберём все потенциальные делители данного числа до его квадратного корня и проверим, делит ли какое-либо число его без остатка. Если такое число есть, то данное число является составным, иначе - простым.

алг Лабиринт

нач

цел  $N, i, j, \text{hor}$ ~~цел таб~~  $a [N, N]$ нц для ввода  $N$ цел таб  $a [N, N]$ ~~нц~~ нц для  $i$  от 0 до  $N-1$  шаг 1:нц для  $j$  от 0 до  $N-1$  шаг 1:ввод  $a [i, j]$ 

кц

если  $\text{mod}(N, 2) = 0$ то  $\text{hor} := 2$ иначе  $\text{hor} := 3$ 

все

лит ~~на~~  $\text{dir} = \text{"вверх"}$ ~~нц~~ ~~нц~~ цел  $x, y$  $x := N - 1$  $y := N - 1$ нц пока  $x < 0$  и  $y < 0$ если  $y + 1 \geq N$  или  $y - 1 < 0$ 

то |двигаемся по горизонтали

нц ~~нц~~  $\text{hor}$  разпростое\_ли( $a[y, x]$ ),  $y, x$ ) $x := x - 1$ 

кц

~~иначе~~ / ~~и~~ если  $\text{dir} = \text{"вверх"}$  тото  $\text{dir} := \text{"вниз"}$ иначе  $\text{dir} = \text{"вверх"}$ 

все

иначе



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N4 (продолжение)

если dir = "вверх"

то

простое\_ли(a[y, x], y, x)

~~y := y + 1~~ y := y - 1

иначе если dir = "вниз"

то

простое\_ли(a[y, x], y, x)

y := y + 1

все

.. все

все

кц  
кон

алг простое\_ли(цел n, цел y, цел x)

нач

лог простое

простое := да

~~цел~~ цел d

цц для d от 2 до sqrt(n) шаг 1

если mod(n, d) = 0 то

простое := нет

.. выход

все

кц

если простое = да

то вывод "Повыбираем яблочко на клетке (" , <sup>x</sup> ~~y~~+1, " , " , <sup>y</sup> ~~x~~+1, ")

иначе вывод "Оставляем яблочко на клетке (" , x+1, " , " , y+1, ")

.. все

кон





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N5.

Перед началом проверки необходимо упорядочить таблицу точек по возрастанию значения  $x$  точки. Если значение  $x$  меньше какого либо другого  $x$ , то пара  $(x, y)$  будет находиться в таблице первой. Далее нужно будет перебрать все данные значения  $x$  в упорядоченной таблице, и, если для следующего большего значения  $x$  координата  $y$  строго меньше, чем  $y$  рассматриваемого  $x$ , то переходим к следующему элементу таблицы, но в ином случае функция будет не монотонно убывающей.

Итоговая асимптотика алгоритма будет зависеть от быстроты функции сортировки. Для очень больших  $N$  можно будет реализовать сортировку ставками, имеющую асимптотику  $O(N \log_2 N)$ , что будет достаточно быстрым решением.

В приведённом алгоритме для простоты применяется сортировка пузырьком, имеющая сложность  $O(N^2)$ .

алг Проверка - функции

нач

цел  $N, i$ ввод  $N$ вещ таб  $f[2, N] \mid f[0, i] - x_i, f[1, i] - y_i$ нц для  $i$  от 0 до  $N-1$  шаг 1ввод  $f[0, i], f[1, i]$ 

кц

сорт( $f$ ,  $N$ )~~нц для  $i$  от 0 до  $N-1$  шаг 1~~лог ~~убывает~~ ans

ans := Да

нц для  $i$  от 0 до  $N-1$  шаг 1если  $i+1 < N$  тоесли  $f[1, i] \geq f[1, i+1]$  то

ans := Нет

.. выход

всё

всё

кц

~~нц~~ если ans = Да

то вывод "функция монотонно убывает"

иначе вывод "функция не является монотонно убывающей"

всё

кц





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N5 (продолжение)

алг сорт (~~вещ~~ арг през вещь таб  $a$ , ~~цел~~  $N$ )

нач

~~вещ~~ цел  $i, j$ ~~вещ~~ цел для  $i$  от 0 до  $N-1$  шаг 1 2~~вещ~~ цел для  $j$  от 0 до  $N-i-1$  шаг 1~~если  $a[j+1] < a[j]$  то~~~~swap ( $a[j]$ ,  $a[j+1]$ )~~если  $a[0][j+1] < a[0][j]$  тоswap ( $a[0][j]$ ,  $a[0][j+1]$ )swap ( $a[1][j]$ ,  $a[1][j+1]$ )

всё

всё  $K_{i1}$   $K_{i2}$ 

Где swap (~~вещ~~  $a$ ,  $b$ ) - функция обмена значений переменных  $a$  и  $b$

*реализация?*

## Олимпиада школьников «Надежда энергетики»

IIIF03	Дистанционно с использованием ВКс
--------	--------------------------------------

№ группы

Место проведения

GR 14-15
----------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № \_\_\_\_\_

ФАМИЛИЯ СКРЯБИН

ИМЯ ИВАН

ОТЧЕСТВО АЛЕКСАНДРОВИЧ

Дата рождения 01.07.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на \_\_\_\_\_ листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады: (ио)

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверляется только то, что записано с этой стороны листа в рамке справа

Задание 12.

Будем считать, что данная формула справедлива с точностью до  $0,0001 = 10^{-4}$ , если число

$a$ , полученное по этой формуле удовлетв.:

$$\begin{cases} a^2 \leq 19 \leq (a+10^{-4})^2 \\ (a-10^{-4})^2 \leq 19 \leq a^2 \end{cases} \Leftrightarrow (a-10^{-4})^2 \leq 19 \leq (a+10^{-4})^2$$

Для того, чтобы найти число  $a$ , будем использовать след. алгоритм:

0) зададим  $d$  - глубину расчёта

если  $d=0$ , то  $a=4$

$d=1$ ,  $a = 4 + \frac{1}{2}$

$d=2$ ,  $a = 4 + \frac{1}{2 + \frac{1}{3}}$

...

1) Последовательно будем считать число для глубины  $d$  и  $d+1$ , пока  $|a(d+1) - a(d)| \geq 0,0001$

(т.е. начнём считать с  $a(1) - a(0)$ , если нет, то

$a(2) - a(1)$  и т.д.

т.е. увеличиваем  $d$  на 1 пока не добьёмся  $|a(d+1) - a(d)| < 10^{-4}$ )

Реализация алгоритма:

Вещ

Форм

$a$  (цел  $d$ , цел  $k$ , цел  $arr$ , цел  $size$ ):

если  $d \leq 1$ :

вернуть  $a(d-1) / (arr[k])$ ;  $\swarrow$   $k \% size$

иначе:

вернуть  $1 / (arr[k] + a(d-1, k+1, arr, size))$ ;  $\swarrow$   $k \% size$

// Функция считает число  $a$ ; от глубины  $d$ , и  $/$  - оператор вещественного деления, а на вход надо подать  $arr = \{2, 1, 3, 1, 2, 8\}$   $size = 6$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Дальнейший код используется реализацией  
 $a(j, k, arr, size)$  :

АЛГ Задача N2  
 НАЧ

ЦЕЛ  $j = 0$ ;

ЦЕЛ  $arr = \{2, 1, 8, 1, 2, 8\}$ ;

ЦЕЛ  $prev = 4$ ;

ЦЕЛ  $cur = 4 + a(j+1, 0, arr, 6)$ ;

ПОКА  $cur - prev \geq 0.0001$  :

$j = j + 1$ ;

$prev = cur$ ;

$cur = a(j+1, 0, arr, 6)$ ;

ЕСЛИ  $(cur - 0.0001) * (cur - 0.0001) < 19 \leq (cur + 0.0001) * (cur + 0.0001)$ ;

ВЫВЕСТИ („СПРАВЕДЛИВА“);

ИНАЧЕ :

ВЫВЕСТИ („НЕТ“);

КОН

Задача N4.

Пусть левая нижняя клетка имеет координаты  $(0; 0)$ , правая верхняя -  $(N-1; N-1)$

Алгоритм проверки шихки:

Для того, чтобы проверить, что число  $P$  является простым ~~нужно~~ достаточно проверить являются ли числа ~~от~~  $k$  от 2 до  $\sqrt{P}$  делителями:

т.е.  $\forall k \in [2; \sqrt{P}] \cap \mathbb{Z} : P \not\equiv 0 \pmod{k}$ .

Т.о.:

1. Для ~~каждого~~ каждого целого  $k$  от 2 до  $\sqrt{P}$ , если  $P \% k == 0$ , то  $P$  - не простое ~~число~~  
 $\Rightarrow$  не берём шихку, заканчиваем алгоритм.

2. Если  $P \% k \neq 0 \forall k \in [2; \sqrt{P}] \cap \mathbb{Z}$ , то  $P$  - простое, шихку берём.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Алгоритм прохождения:

0. Стартуем в  $(0; 0)$ , идём  $(N // 2)$  шагов вверх, выполняю для каждого шага алгоритм проверки шашки. Оказываемся в точке  $(0; N // 2)$ .

1. Идём  $(N-1)$  шагов вправо, выполняя для каждого шага алгоритм проверки шашки. Оказываемся в т.  $(N-1; N // 2)$

2. Идём  $(N // 2 - (N+1) \% 2)$  шагов вверх, выполняя для каждого алгоритм проверки шашки, Оказываемся в т.  $(N-1; N // 2 + N // 2 - (N+1) \% 2)$

Докажем, что  $\forall N \in \mathbb{N}$

$$N // 2 + N // 2 - (N+1) \% 2 = N-1$$

"%" и "//" - оператор целочисленного деления.

Если  $N = 2k+1$ , то  $N // 2 = k \Rightarrow$

$$N // 2 + N // 2 = 2k$$

$$(2k+1+1) \% 2 = 0 \Rightarrow N // 2 + N // 2 - (N+1) \% 2 = 2k - 0 = 2k = N-1$$

Если  $N = 2k$ , то  $N // 2 = k$

$$(2k+1) \% 2 = 1 \Rightarrow N // 2 + N // 2 - (N+1) \% 2 = k + k - 1 = 2k - 1 = N-1$$

ч.т.д.

Тогда в конце мы оказываемся в т.  $(N-1; N-1)$ , что и является целью.

P.S. В данном решении если  $N \equiv 0 \pmod{2}$ , то схема выглядит так:



Если она выглядит так:



то, в п.0 и п.2 надо поменять кол-во шагов местами



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Задача 15.

Ф-ция  $f(x)$  является монотонно возрастающей, тогда и только тогда, когда  $\forall x_1, x_2 \in E(f)$  если  $x_1 > x_2$ , то  $f(x_1) > f(x_2)$ .

Тогда если мы отсортируем таблицу по возрастанию аргумента, т.е.  $\forall n \in \mathbb{N}$ , причём  $n > n_0$ , где  $n_0$  - номер рассматриваемой точки,  $x_n > x_{n_0}$ , тогда по опр.  $y_n > y_{n_0}$ , если  $f(x) \nearrow$  на  $E(f)$ .

Тогда после сортировки ~~необязательно~~ достаточно проверить возрастают ли значения в таблице.

Для сортировки будем использовать алгоритм «быстрой сортировки» («quick sort»). Его описание:

0. Если раз-во рассм. эл-ов равно 1, то область отсортирована, дальше не идём.

1. Устанавливаем опорный эл-т, пусть это будет центральная, по индексу, пара. ~~или~~

2. Располагаем все пары, у которых аргумент меньше чем аргумент опорной пары слева от неё, у которых аргумент больше, либо равен - справа от опорной пары.

3. Применяем п.0, 1, 2, 3 для пар, стоящих слева от опорного, и для пар, стоящих справа от опорного. (Применяем по отдельности, т.е. фраза в п.0 «дальше не идём», касается только левой или правой области).



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

После того, как пары отсортированы по возрастанию аргумента  $x$ , начнём проверку на возрастание значений:

0. Число  $n=0$

1. Если число  $n \geq N-1$ , то ~~обработка~~ заканчиваем алгоритм, говорим, что  $f(x)$  возрастает.

2. Смотрим на пару под номером  $n$  и  $n+1$ , если на  $y$  пары  $n+1$ , значение больше чем у  $n$ , то увеличиваем  $n$  на 1 и выполняем п.1 и п.2. Если у пары  $n+1$  значение меньше или равно значению пары  $n$ , то ~~выводим~~ говорим, что  $f(x)$  не возрастает и заканчиваем алгоритм.

Задача 13.

1.  $A \rightarrow C = 1, C = 0$

$A \rightarrow C = \neg A \vee C$ , т.к.  $C = 0$ , и  $\neg A \vee C = 1$ , то  $\neg A = 1 \Rightarrow A = 0$

2.  $\neg B \rightarrow E = 1, E = 0$

$\neg B \rightarrow E = \neg(\neg B) \vee E = B \vee E$ , т.к.  $B \vee E = 1$  и  $E = 0$ , то  $B = 1$

3.  $\forall A \exists B \rightarrow \exists B$

Докажем, что  $\forall A \rightarrow B = \neg A \vee B$ , при помощи таблицы истинности:

A	B	$\neg A$	$\neg A \vee B$	$A \rightarrow B$
1	1	0	1	1
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

Ответ:  $A=0$

$B=1$





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Задание 13.

1. Пусть  $x$  и  $y$  - координаты робота, левый нижний угол -  $(0; 0)$ , правый верхний -  $(M-1; N-1)$ .

1.  $x=0$ ,  $y=0$  изначально.

2. Если  ~~$y+3 < N$~~ :

Если  ~~$x+2 < M$~~ :

Если от  $y$  до  $y+3$  при ~~каждом  $x$~~ ,  
нет ~~штук~~ и на ~~клетке  $(x+1, y+1)$~~



2. Если  $x == M-1$  и  $y == N-1$ , то путь сузу.  $\Rightarrow$  можно, алгоритм завершаем полностью.

3. Если  ~~$y+3 < N$~~ , то ~~и  $x+2 < M$~~

Введём обозначение условий: (1)  $y+3 < N$  (2)  $x+2 < M$   
(3)  $x+1 < M$  (4)  $x+1 < N$   
(5)  $x-1 \geq 0$  (6)  $x-2 \geq 0$   
~~(7)  $y+1 < N$~~

4. Если (4) и (3)

то и на ~~всех~~ клетках  $(x, y)$  и  $(x, y+1)$  нет штыка,  
то если на ~~какой-то~~ кл.  $(x, y)$ ,  $(x+1, y+1)$  или  $(x+2, y+1)$   
или  $(x+3, y+1)$   
находится штык, то берём наименьшую  
по аргументу клетку с штыком из этих и  
применяем п. 2 и последующие.

Иначе рассм. все штыки по очереди  
от  $(x+3, y+1)$ , до  $(x+2, 0)$  (уменьшаем  $y$  на 1),  
если на клетке штык то применяем для  
ней п. 2, дальше кл. от  $(x+3, y+1)$  до  $(x+2, 0)$  не  
рассм. (все ~~применяется~~ ~~невозможно~~ ~~не~~).



ВНИМАНИЕ! Проверляется только то, что записано с этой стороны листа в рамке справа

5. Если (4) и (6)

и на кл.  $(x; y+1)$  нет плитки, то

~~если~~ на кл. рассм. кл. от  $(x; y+1)$  до  $(x-3; y+1)$  по очереди, если на какой-либо из них плитка, то для неё прищ. п.2 и ~~еще~~ остальные кл. не рассм., если такой кл. не нашлось рассм. кл. от  $(x+3; y+1)$  до  $(x-3; 0)$ , ~~то~~ по очереди, если на клетке плитка, то для неё п.2 и остальные не рассм.

6. Если (1) и (2) и на кл.  $(x; y+1)$   $(x; y+2)$   $(x; y+3)$

нет плиток, то ~~и если на кл.  $(x$~~

то рассм. кл. от  $(x+3; y+3)$  до  $(x+3; 0)$  по очереди, если на кл. плитка, то для неё п.2. и остальные не смотрим.

7. Если (1) и (5) и на кл.  $(x; y+1)$   $(x; y+2)$   $(x; y+3)$

нет плиток, то рассм. кл. от  $(x-1; y+3)$  до  $(x-1; 0)$  по очереди, если на кл. плитка, то для неё п.2 и остальные не смотрим.

Если алгоритм завершился и мы не скажем, что возможно добраться, то добраться невозможно

Олимпиада школьников «Надежда энергетики»

11 F03	Олимпиада школьников и школь- зователей ВЭС
--------	--

№ группы

Место проведения

GR14-89
---------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № \_\_\_\_\_

ФАМИЛИЯ СТРИГАЛЕВ  
ИМЯ НИКИТА  
ОТЧЕСТВО СЕРГЕЕВИЧ

Дата рождения 04.12.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 08 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:

С. Стригалев

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

к1.

Рассмотрим как первое утверждение:  $A \rightarrow C = 1$ . Мы знаем что  $C = D = E = 0 \Rightarrow$  перепишем утв. в виде:  $A \rightarrow 0 = 1 \Rightarrow \Rightarrow A = 0$ , т.к. при  $A = 1$  получим  $1 \rightarrow 0 = 0$  - противоречит условию. Рассмотрим как 3 утв.:  $\neg B \rightarrow E = 1$ . Перепишем в виде:  $\neg B \rightarrow 0 = 1 \Rightarrow \neg B = 0$ , т.к. при  $\neg B = 1$  получим  $1 \rightarrow 0 = 0$  - противоречит условию  $\Rightarrow \neg B = 0 \Rightarrow B = 1$ . Получим  $A = 0, B = 1$ . Подставим во второе утверждение:  $A \wedge B \rightarrow D = 1; 0 \wedge 1 \rightarrow 0 = 1; 0 \rightarrow 0 = 1; 1 = 1$  - верно.  $\oplus$

Ответ:  $A = 0; B = 1$ .

к2.

arr модуль (arr вещ  $\alpha$ )

ком

если  $\alpha < 0.0$  то  
вернуть  $-\alpha$

всё

если  $\alpha \geq 0.0$  то  
вернуть  $\alpha$

всё

ком

arr f (arr вещ  $\alpha$ )

ком

вещ  $\alpha \wedge S = P.0 + X$

вещ  $\alpha \wedge [5]$

$\alpha \wedge [0] = 2$

$\alpha \wedge [1] = 1$

$\alpha \wedge [2] = 3$

$\alpha \wedge [3] = 1$

$\alpha \wedge [4] = 2$

к3 - ком

(продукты. на шестерке)



ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа

н) (продолжение)

для  $i$  от 0 до 4

нц

$$ans = 1.0 / ans + a[i]$$

кц

вернуть ans

кон

для рамки

нов

все res

$$res = \text{sqrt}(19)$$

$$\text{все prev} = 0.0$$

все check

$$\text{check} = f(0.0)$$

пока  $(\text{check} - \text{prev}) > 0.0001$

нц

$$\text{prev} = \text{check}$$

$$\text{check} = f(1.0 / \text{check})$$

кц

$$\text{check} = 1.0 / \text{check} + 4$$

если  $|\text{check} - \text{res}| \leq 0.0001$  то

вывод 'Верно'

иначе

вывод 'Неверно'

всё

кон

Для рамки решает задачу. Примечание:  
В строке for для  $i$  от 0 до 4 значения индексов  
нельзя, т.е. не  $i=0, i=1, i=2, i=3, i=4$ ;  $\text{sqrt}(19)$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№ 5.  
алг. QuickSort (выр. пог. цел.  $x[n] \dots [2]$ , ам. цел. left, ам. цел. right)

кар.

цел.  $i, j, k, y$

$$k = x[(left + right) div 2][0]$$

$$i = left$$

$$j = right$$

пов. порядок

пока  $x[i][0] < k$

цел.

$$i = i + 1$$

цел.

пока  $x[j][0] > k$

цел.

$$j = j - 1$$

цел.

если  $i < j$  то

$$y = x[i][0]$$

$$x[i][0] = x[j][0]$$

$$x[j][0] = y$$

$$y = x[i][1]$$

$$x[i][1] = x[j][1]$$

$$x[j][1] = y$$

$$i = i + 1$$

$$j = j - 1$$

(проверка только на элемент  $y$ )



ВНИМАНИЕ! Проверяется только то, что записано в этой стороне листа в рамке справа

n5 (программа)

инкре

если  $i = j$  то

$i = i + 1$

$j = j - 1$

else

if  $i > j$

если left < j то

QuickSort(x, left, j)

если i < right то

QuickSort(x, i, right)

ком

алг. Меноматрица

ком

цел n

блог n

цел a[n][2]

для i от 0 до n-1

мы

блог a[i][0], a[i][1]

мы

QuickSort(a, 0, n-1)

для i от 1 до n-1

мы

~~мы~~

(программа на листе 5)



ВНИМАНИЕ! Проверяется только то, что записано  
с левой стороны листа в рамке справа

15 (продолжение)

если  $a[i-1][1] \neq a[i][1]$  то

вывод 'Не монотонно возрастающая'  
щодя'

вывод 0

всё



иц

вывод 'Монотонно возрастающая'

иц

Алгоритм таков: сортируем массив по возрастанию  $x$  с помощью быстрой сортировки, а потом проверяем все значения  $y$  и если какое-то имеет предположительно значение не меньше предыдущего, то  $arr[x]$  не монотонно возр.

Прим.: Вывод 0 означает вывод из программы (дополное завершение) и значения в массиве для вкл. и выкл. элементов.

иц

arr[15-primе (arr[15] и 1)

иц

если  $x < 2$  то

вернуть 0

всё

~~иц~~  
~~вернуть 0~~

иц

(представление на листе 6)



ВНИМАНИЕ! Прочитывается только то, что записано с этой стороны листа в рамке справа

нц (продолжение)

если  $x = 2$  то

вернуть истину  
всё

если  $x \bmod 2 = 0$  то

вернуть ложь

всё

для  $i$  от 3 до  $\sqrt{x}$  (при  $x \in \mathbb{N}$ ) с шагом 2

нц

если  $x \bmod i = 0$  то

вернуть ложь

всё

нц

вернуть истину

кон

если  $\sqrt{x}$  не целое

кон

цел  $n$

$\forall i \in [1, n]$

цел  $a[i] \in [1, n]$

для  $i$  от 0 до  $n-1$

нц

для  $j$  от 0 до  $n-1$

нц

$\forall i \in [1, n]$

нц

нц

цел count

count = 0

проверить, не лишний ли?



ВНИМАНИЕ! Проверять только то, что записано  
→  
с этой стороны листа в рамке справа

и ч (продолжить)

для collected[с.к.]

ног f = штижа

для i = n-1 до 0 с шагом -2

нз

если f то

для j = n-1 до 0 с шагом -1

нз

если is\_prime(a[i][j]) то

collected[count] = a[i][j]

count = count + 1

всё

нз

~~нз~~

иначе

для j = 0 до n-1

нз

если is\_prime(a[i][j]) то

collected[count] = a[i][j]

count = count + 1

всё

нз

всё

если f = штижа то

f = лоты

иначе

f = штижа.

всё

нз

Продолжи. на листе 8/



ВНИМАНИЕ! Проверьте, чтобы запись  
с этой стороны была в рамке справа

```

микропрограмме!
вывод 'Собрано изделий:'
вывод count
вывод '\n'
для i = 0 до count - 1
  выв
  вывод collected
  вывод " "

```



кон  
 В условии задачи и в примере даны  
 разные направления движения, пример  
 за верное то, которое показано на рисунке  
 В задаче предполагается, что: 1) Вывод  
 не выводит много, вроде того, что вы  
 дано ('\n' - знак переноса строки) 2) Вывод  
 для значения выводится.

Олимпиада школьников «Надежда энергетики»

110701 | Институционно, с  
использованием ВКС  
№ группы | Место проведения

QV 97-99  
шифр

— Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73101

ФАМИЛИЯ СТРУГОВ

ИМЯ АЛЕКСЕЙ

ОТЧЕСТВО ЮРЬЕВИЧ

Дата рождения 12.10.2004

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 5 листах

Дата выполнения работы: 28.02.21  
число, месяц, год.

Подпись участника олимпиады:

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Дано  
 $A \rightarrow C$   
 $A \& B \rightarrow D$   
 $\neg B \rightarrow E$   
 $C=0$   
 $D=0$   
 $E=0$

Решение

①  $A \rightarrow 0 \Rightarrow A=0$   
 ②  $\neg B \rightarrow 0 \Rightarrow \neg B=0 \Rightarrow B=1$   
 ③  $A \& B \rightarrow D \Rightarrow \begin{cases} B=0 \\ B=1 \end{cases}$   
 ④  $\begin{cases} A=0 \\ B=0 \\ B=1 \\ B=1 \end{cases} \Rightarrow A=0, B=1$



Ответ: A - ложно, B - истинно

~~$n = \text{ввод}(\text{"ВВЕДИТЕ n: "})$   
 $\text{нц. пока } i = 0$   
 $j = 0$   
 $a = 1$   
 $\text{нц. пока } i < n:$   
 $\text{нц. пока } j < n:$   
 $a[i][j] = \text{ввод}(\text{"ВВЕДИТЕ ЧИСЛО"})$   
 $j = j + 1$   
 $i = i + 1$   
 $i = n - 1$   
 $j = n - 1$   
 $\text{нц. пока } i \neq 0 \text{ и } j \neq 0:$   
 $\text{нц. пока } i \neq 0:$~~



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

    ~4
    ФУНКЦИЯ ПРОСТ(K):
    НУ ПОКА L ≤ ceil(K/2) - 1.
    ЕСЛИ K % L ≠ 0:
        return True.
    ИНАЧЕ:
        L = L + 1
    КУ.
    return False.
  
```

```

    n = ВВОД("ВВЕДИТЕ n:")
    j = 0
    i = 0
    a = []
    НУ ПОКА i < n:
        НУ ПОКА j < n:
            a[i][j] = ВВОД("ВВЕДИТЕ число.")
            j = j + 1
        КУ.
        i = i + 1
  
```

Проверка N

7

```

    КУ.
    i = n - 1
    j = n - 1
    НУ ПОКА i ≠ 0 и j ≠ 0:
        НУ ПОКА i ≠ 0:
            ЕСЛИ a[i][j] ПРОСТ(a[i][j]):
                ВЫВОД(a[i][j])
            ЕСЛИ i = 0 и j = 0: КОНЕЦ.
            НУ ПОКА ЕСЛИ ПРОСТ(a[i][j-1]):
                ВЫВОД(a[i][j-1])
            j = j - 1
        НУ ПОКА i ≠ n:
            ЕСЛИ ПРОСТ(a[i][j]):
                ВЫВОД(a[i][j])
            i = i + 1
    i = n - 1
    ЕСЛИ ПРОСТ(a[i][j-1]):
        ВЫВОД(a[i][j-1])
  
```



QV 97-99



ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамках спрэда

$j = \leftarrow j - 2$  <sup>n-4 (продолжение)</sup>  
кч.

$a = \text{КОР}(?)$ , точки  $(a \cdot 10^{-4})$  <sup>n-2</sup>

~~ПОКА~~ True:

ЕСЛИ  $a > b$ :

КОНЕЦ ВЫВОДА ("true")

$b = 1 / (1 + (1 / (1 + (1 / (1 + \dots))))$

$(1 / (4 + b))$ , точки  $(10^{-5})$

ЕСЛИ  $a > 2 + b$ :

ВЫВОД ("False")

КОНЕЦ

кч.

$a = [5]; b = [5]$  <sup>n-5</sup>

$n = \text{ВВОД}()$

$i = 0; j = 0$

ПОКА  $i < n$ :

$a[i] = \text{ВВОД}()$  ("a")

$b[i] = \text{ВВОД}()$  ("b")

$i = 0$   
ПОКА  $i < n$ :

пока  $j < (n - i) + 1$

$a[j],$  ЕСЛИ  $a[j] > a[j+1]$ :

$a[j], a[j+1] = a[j+1], a[j]$

$b[j], b[j+1] = b[j+1], b[j]$

$i = 0$   
 $j = 0$

С  
О  
Д  
Р  
Ж  
А  
Н  
И  
Е



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

```

k = True ~ 5 (продоление)
ПОКА i < N-1:
    ЕСЛИ и  $\{l[i] \leq \{l[i+1]\}$ :
        ВЫВОД ("False") k = False
    КОНЕЦ
ВЫВОД (k)

```



Решение: Известно, что даны два массива чисел. Одно значение  $x$ , другое  $y$ . Значение  $x$  мы сортируем первый массив по возрастанию, а второе  $y$  также в соответствии со значениями  $x$ . Мы получим отсортированный массив  $x$  значений и массив  $y$  соответствующий  $x$ . Проверим если  $y$  отсортирован так, что значения идут от наибольшего к наименьшему, то функция монотонно убывает, иначе это нет.

```

i = 0 j = 0 13
M = ВЫВОД ( ) %
N = ВЫВОД ( )
ЕСЛИ (M+N) % 3  $(M-N) \geq M \% 4 + N \% 3$ :
    ВЫВОД ("False")
ПОКА i < M:
    ПОКА j < N:
        k = ВЫВОД ("Совпадают ли КЛАВИА?")

```





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

13 (предположение)

ЕСЛИ  $k = 1$ ;  
 $x = k \cdot a[i] = i$   
 $y = j \cdot b[j] = j$   
 $i = n$   
 $j = 0$

$T = true$   
ПОКА  $i \neq 0$  и  $j \neq n$ ;

ЕСЛИ  
ПОКА  $i < \text{лин}(a)$  или  $a[i] < j+1$  и  
или  $(b[j] > i+1$  или  $(b[j] < i-1)$ ;

$i = i - 1 \rightarrow$   
 $j = j + 1 \rightarrow$

ПОКА  $i < \text{лин}(a)$  или  $a[i] < j+1$  и  
или  $(b[j] > i-3$  или  $(b[j] < i-3)$ ;

$i = i - 3$

$j = j + 1$

ИНАЧЕ ЕСЛИ  $(a[i] > j - 1$  или  $a[i] < j - 1)$  и  
или  $(b[j] > i + 1$  или  $b[j] < i + 1$

$i = i + 1$

$j = j - 1$

ИНАЧЕ ЕСЛИ  $(a[i] > j - 1$  или  $a[i] < j - 1)$  и  
или  $(b[j] > i + 3$  или  $b[j] < i + 3$

$i = i + 3$

$j = j - 1$

ИНАЧЕ:  
 ~~$T = false$~~   $T = false$   
Конец

ВЫВОД(T)

# Олимпиада школьников «Надежда энергетики»

IGF01	Олимпиада школьников с использованием ВКС
№ группы	Место проведения

VO 54-41
шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73991

ФАМИЛИЯ Тимофеев

ИМЯ Григорий

ОТЧЕСТВО Михайлович

Дата рождения 18.08.2004

Класс: 9

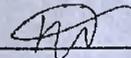
Предмет информатика

Этап: заключительный

Работа выполнена на 3 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

$$N2 \quad N=64$$

$$P=256$$

$$\text{цел } M[10002]=0$$

$$M[0] = N \% 10$$

$$M[1] = N / 10$$

для  $i$  от 1 до  $P$

для  $j$  от 0 до 999

$$M[i] = M[i] \cdot N$$

конеч-ья

для  $j$  от 0 до 998

$$M[j+1] = M[j+1] + M[j] / 10$$

$$M[j] = M[j] \% 10$$

конеч-ья

конеч-ья

для  $I$  от 0 до  $P-1$

если  $M[I] = 8$ , то вывести  $(I+1)$   
конеч-ья.

M1 для  $x$  от 0 до 9

для  $y$  от 1 до 9

для  $z$  от 0 до 9

$$\text{если } (20010002022004 + 100010000100 * x + \\ + z * 100000000 + 100000010 * y; 100 * y + x + z) = 0, \text{ то}$$

вывести  $(x, y, z)$

конеч-ья

конеч-ья

конеч-ья

конеч-ья



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№3 Вывод S // 16-тиричное число строк

$n = 0$   
 для  $i$  от 0 до длины  $S$ -1  
 если  $(S[i] > 'a')$  и  $(S[i] < 'z')$ , то  
 $S[i] = S[i] - \text{key}('a') + \text{key}('A')$   
 концы-если  
 концы-где

для  $i$  от 0 до длины  $S$ -1  
 если  $S[i] > '0'$  и  $S[i] < '9'$ , то  
 $n = n * 16 + (\text{key}(S[i]) - \text{key}('0'))$   
 иначе  
 $n = n * 16 + (\text{key}(S[i]) - \text{key}('A')) + 10$   
 концы-где  
 $S = "$  "

пока  $n > 0$  ~~?~~  
 $x = n \% 10$   
 $S1 = "$  "  
 для  $i$  от 1 до  $x$   $S1 = S1 + "$  "  
 $S = "(" + S1 + ")" + S$   
 $n = n / 10$   
 концы-пока  
 Вывод S.



№5. (1)  $A \rightarrow B$   
 (2)  $A \& B \rightarrow D$   
 (3)  $\neg B \rightarrow E$

} истинны C, D, E - ложно  
 A, B - ?

A - ложно, т.к. из истинны не следует ложь в (1)  
 (3)  $\neg B$  - ложно аналогично (1), B - истинно  
 (2)  $0 \& 1 \rightarrow 0$  - истинно



Символы: A=0; B=1



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

$$N4 \quad \sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \dots}}$$

$$\epsilon = 0,0001$$

$$e = \text{sqrt}(2)$$

$$p = 0,5$$

$$it = 1$$

пока  $it \leq 1000000$

$$p = 1 / (2 + p)$$

$$; it = it + 1$$

если  $abs(1 + p - e) < \epsilon$

вывод из цикла

концы - если

концы - пока

если  $it \geq 1000000$ , то

вывод "Гипотеза не верна"

иначе

вывод "Гипотеза верна, итераций = ",  $it$ .



I11F03	Дистанционно с использованием ВКС
--------	--------------------------------------

№ группы

Место проведения

GR 14-10
----------

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

Вариант № 73 111

ФАМИЛИЯ ХАРЛУНИН

ИМЯ АЛЕКСАНДР

ОТЧЕСТВО АЛЕКСАНДРОВИЧ

Дата рождения 31.07.2003

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 6 листах

Дата выполнения работы: 28.02.2021  
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N 1.

Упростим данные выражения:

1)  $A \rightarrow C = \bar{A} + C = 1$

2)  $A \& B \rightarrow D = \overline{(A \& B)} + D = \bar{A} + \bar{B} + D = 1$  (закон де Моргана)

3)  $\bar{B} \rightarrow E = B + E = 1$

$C = D = E = 0$  (по условию)

A = 0 (что бы 1 выражение было истинно)

рассмотрим второе выражение:

$1 + \bar{B} + 0 = 1$

B может быть как 1, так и 0

рассмотрим третье выражение:

$B + 0 = 1$

$B = 1$  (что бы 3 выражение было истинно)

Итого выполниме истинно или ложно для A и B

Ответ: A - ложно; B - истинно.

N 2.

 $\sqrt{19}$  - константа, запишем ее в переменную left side (вещное число)

{псевдокод}

left side = sqrt(19) const float

переменная right side - изменяется, ее мы будем

изменять при помощи цикла while

условие цикла while:

while  $|\text{left side} - \text{right side}| \geq 0.0001$

(пока выполняем по максимуму разницы до тех пор, пока не достигнем)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

ПРОДОЛЖЕНИЕ №2

структура из данных чисел, олицетворяющих  
элементы, будет записана в виде кортежа  
(неизменяемого списка), кратчайшего пути числа  
structure = tuple of int (2,1,3,1,2,8) const

len = 6

значения будут поочередно входить по индексу  
если i (индекс) равен 6, а цикл еще идет, i тогда  
становится 0

по умолчанию i = 0 (int), но с помощью итерации  
цикла while i прибавляет к себе единицу

structure\_first = 4 int const.

суть алгоритма:

считают индекс числа будет с конца, так как  
первое число структуры умножится, на индекс  
зачислит его в значение

полный вариант системы исчисления

программа задание 2.

left side = sqrt(19) const float

right side = 0 float

structure = tuple of int (2,1,3,1,2,8) const

len = 6

i = 0

tochnost = 4.0001

structure.index(i)

right side = structure[i]

i = i + 1

пока |left side - right side| > tochnost break:

right side = structure[i] + 1 / right side

i = i + 1

если i = 6 тогда i = 0

вывод "истина"



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

(интерпретация №2)

применение к коэф:

если вычисление инициализация = программа уже введена.

если вычисление не инициализация = программа не инициализирована и инициализация  
сущности инициализация

ответы.

№5

функция является монотонно возрастающей, значит она возрастает по всей осях, значит можно только проверить все значения функции.

в данном формате решения - проверить значения функции в диапазоне на столе (сидя) (уже отсортированные по возрастанию значения  $x$ , значения  $y$  проверять по значениям  $x$  (значения  $y$  являются константой)

программа работает - б  
подключить таблицу table  
вход  $N$   
 $y = previous - 1$ . бесконечность  
для  $i$  от 1 до  $N$  делать:  
 $x = table[i]$   
 $y = table[x]$   
если  $y \leq previous$  то выход  
иначе:  
выход  $i$ , не возрастает.  
конечу  
выход «возрастает»

использование:  
конструкция  $y = table[x]$   
используется, если  
в выводе для преобразования  
выход.





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

нч.

Итак, нам объявлены функции, определяющие функцию  $easy$ .

функция  $easy(n: int): bool$

count = 0 int

k = 2 int

пока  $(k * k \leq n)$  и  $(count == 0)$  делать:

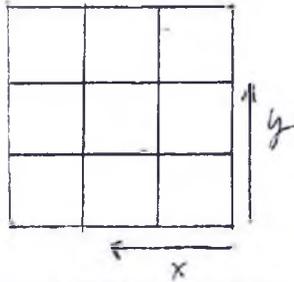
если  $n \bmod k == 0$  тогда count = count + 1

k = k + 1

если count == 0 выдать True

иначе выдать False

В программе объявлены следующие тестовые функции как глобальные массивы, в которых мы получаем число по 2-м координатам  $(x, y)$



примечание: условия тестирования рисунка, соответное как на рисунке.

программа задание 4  
пожалуйста, чтобы таблица Table  
выдать N

summa = 0 int

x = 1 int

y = 1 int

пока  $y < N$  делать:

для  $i$  от 1 до N делать:

если  $easy(table[i][y])$  то summa = summa + table[i][y]

если  $y + 1 < N$  то  $y = y + 1$

если  $easy(table[x][y])$  то summa = summa + table[x][y]

имеем выходы

если  $y + 1 < N$  то  $y = y + 1$

если  $easy(table[x][y])$  то summa = summa + table[x][y]

имеем выходы



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

продолжение проверки, и т.д.

```

для i от n по 1 шагом:
    если easy(табл[i][y]) то сумма = сумма + табл[i][y]
    x = i
    если y + 1 < n то y = y + 1
    [ иначе y = 0
      если easy(табл[x][y]) то сумма = сумма + табл[x][y]
    ]
    [ иначе y = 0
      если y + 1 < n то y = y + 1
      [ иначе y = 0
        если easy(табл[x][y]) то сумма = сумма + табл[x][y]
      ]
    ]
известна сумма
  
```



n 3

большинство программ не имеют представлений о действительности как о непрерывной линии, в которых две величины значимые bool (True - истина, False - ложь)

	F	F	F	T
cn)	F	F	F	F
y	T	F	F	F
	x (m)			

с значениями проставлены для переменных



матрицы и функции, которые возвращают значение bool в зависимости от их аргументов, возвращают ли значения переменные координаты могут быть вычислены и использованы в матрице (после вычисления)

<p>функция a(x: int, y: int): bool</p> <p>new_x = x new_y = y если табл[new_x][new_y + 1]:     известно False иначе вернуть new_x = x + 3 new_y = y + 1 для i от new_y по 1 шаг:     если табл[new_x][i]:         известно True     x = new_x     y = i известно False</p>	<p>функция b(x: int, y: int): bool</p> <p>new_x = x new_y = y если табл[new_x][new_y + 1] or табл[new_x][new_y + 2] or табл[new_x][new_y]:     известно False иначе вернуть new_x = x + 1 new_y = y + 3 если i от new_y по 1 шаг:     если табл[new_x][i]:         известно True     x = new_x     y = i известно False</p>	<p>функция c(x: int, y: int): bool</p> <p>new_x = x new_y = y если табл[new_x][new_y + 1]:     известно False иначе вернуть new_x = x - 3 new_y = y + 1 y = 1 для i от new_y по 1 шаг:     если табл[new_x][i]:         известно True     x = new_x     y = i известно False</p>	<p>функция d(x: int, y: int): bool</p> <p>new_x = x new_y = y если табл[new_x][new_y + 1] or табл[new_x][new_y + 2] or табл[new_x][new_y + 3]:     известно False иначе вернуть new_x = x + 1 new_y = y + 3 для i от new_y по 1 шаг:     если табл[new_x][i]:         известно True     x = new_x     y = i известно False</p>
--	---	--	--



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

перевод для перевода в 3

```

ПРОГРАММА формат. 3
Позвольте написать TABLE
long m int const
long n int const
x=1 int
y=1 int
while x <= m и y <= n :
    while a(x; y)
        while b(x; y)
            while c(x; y)
                while d(x; y)
                    while e(x; y) "не писать" комм.
        while "писать"

```

примечание: алгоритм, представленный выше, является "неверным" по сути от примитивной <sup>версии</sup> ~~версии~~ оптимальной реализации для конкретной задачи.

