Олимпиада школьников «Надежда энергетики».

ЗАДАЧИ ПО ИНФОРМАТИКЕ

11 класс.

1. В нотариальной конторе «Правдодел» есть сейф. К сожалению, директор забыл код от сейфа. Но директор увлекается математикой и помнит, что код – это совершенное число. Ещё он помнит, что в этом числе было M цифр. Число называется совершенным, если оно равно сумме всех своих делителей, кроме этого числа. Помогите директору ограничить диапазон поиска и разработайте алгоритм, который вычисляет возможные совершенные числа, удовлетворяющие условиям выше.

Решение. Перебираем все возможные числа x, состоящие из M цифр (такие числа лежат в диапазоне от 10^{M-1} до $10^M - 1$). Для каждого числа x проверяем, что оно является совершенным. Для этого просматриваем все возможные делители k от 2 до x / 2, проверяем делится ли x на k, если делится, прибавляем k к сумме делителей (любое число делится на 1, поэтому сумма делителей инициализируется значением 1). Если сумма делителей равна числу x, выводим это число.

```
алг КодОтСейфа
нач
  цел m, x

  ввод m
  если m < 1 то
    вывод 'Некорректное значение'
  иначе
    для x от 10 ^ (m – 1) до 10 ^ m – 1
    нц
      если Совершенное(x) то
        вывод x
      всё
    кц
  всё
кон

алг Совершенное(арг цел x)
нач
  цел k, s

  s = 1
  для k от 2 до x div 2
  нц
    если x mod k = 0 то
      s = s + k
    всё
  кц
  если s = x то
    вернуть истина
  иначе
    вернуть ложь
  всё
кон
```

11 класс.

2. Заданы координаты N точек на плоскости (x_1, y_1), (x_2, y_2), …, (x_N, y_N). Определить координаты минимального прямоугольника, содержащего все эти точки, стороны которого параллельны прямым $y = x$ и $y = –x$. Вычислить площадь получившейся фигуры.

Решение. Для каждой точки надо вычислить коэффициент b_1 уравнения прямой $y = x + b_1$ и коэффициент b_2 уравнения прямой $y = –x + b_2$. Далее надо найти максимальное и минимальное значение коэффициентов b_1 и b_2 по всем точкам. Эти прямые и будут образовывать стороны прямоугольника. Вершины прямоугольника – это точки пересечения этих прямых. Их координаты вычисляются следующим образом:

$x_{v1} = (b_{2\max} - b_{1\max}) / 2$, $y_{v1} = (b_{2\max} + b_{1\max}) / 2$

$x_{v2} = (b_{2\max} - b_{1\min}) / 2$, $y_{v2} = (b_{2\max} + b_{1\min}) / 2$

$x_{v3} = (b_{2\min} - b_{1\max}) / 2$, $y_{v3} = (b_{2\min} + b_{1\max}) / 2$

$x_{v4} = (b_{2\min} - b_{1\min}) / 2, \qquad y_{v4} = (b_{2\min} + b_{1\min}) / 2$

Для вычисления площади фигуры надо найти и перемножить длины сторон, т.е. расстояния между точками 1 и 2 и точками 1 и 3.

```
алг Прямоугольник
нач
  цел n, i
  вещ x[n], y[x], b1[n], b2[n]
  вещ b1min, b1max, b2min, b2max
  вещ xv1, yv1, xv2, yv2, xv3, yv3, xv4, yv4

  ввод n
  если n < 1 то
    вывод 'Некорректное значение'
  иначе
    для i от 1 до n
    нц
      ввод x[i], y[i]
    кц

    для i от 1 до n
    нц
      b1[i] = y[i] – x[i]
      b2[i] = y[i] + x[i]
    кц

    b1min = b1[1]
    b1max = b1[1]
    b2min = b2[1]
    b2max = b2[1]
    для i от 2 до n
    нц
      если b1[i] < b1min то
        b1min = b1[i]
      иначе
        если b1[i] > b1max то
          b1max = b1[i]
        всё
      всё
      если b2[i] < b2min то
        b2min = b1[2]
      иначе
        если b2[i] > b2max то
          b2max = b2[i]
        всё
      всё
    кц

    xv1 = (b2max – b1max) / 2
    yv1 = (b2max + b1max) / 2
    xv2 = (b2max – b1min) / 2
    yv2 = (b2max + b1min) / 2
    xv3 = (b2min – b1max) / 2
    yv3 = (b2min + b1max) / 2
    xv4 = (b2min – b1min) / 2
    yv4 = (b2min + b1min) / 2

    вывод 'Координаты вершин'
    вывод '(', xv1, ', ', yv1, ')'
    вывод '(', xv2, ', ', yv2, ')'
    вывод '(', xv3, ', ', yv3, ')'
    вывод '(', xv4, ', ', yv4, ')'
    вывод 'Площадь прямоугольника - ',
           sqrt(sqr(xv1 - xv2) + sqr(yv1 - yv2)) * sqrt(sqr(xv1 - xv3) + sqr(yv1 - yv3))
  всё
кон
```

9 класс.

3. Даны три стопки карточек, на каждой карточке записано два числа. Эти числа задают координаты точки на плоскости. Необходимо подсчитать, сколько равносторонних треугольников среди троек.

Решение. Пусть (x_1, y_1), (x_2, y_2), (x_3, y_3) – координаты трёх точек. Для каждой из трёх пар точек вычислим расстояние между точками (длины сторон возможного треугольника):

$r_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, $r_2 = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}$, $r_3 = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$. Чтобы треугольник был равносторонним, должно выполнять условие $(r_1 = r_2)$ & $(r_1 = r_3)$ (тогда $r_2 = r_3$).

Для всех троек вычисляем длины сторон, проверяем выполнение условия. Также надо проверить, что треугольник вообще существует. Длины сторон по определению неотрицательны, но надо проверить, что они строго больше 0. Также каждая сторона должна быть меньше суммы двух других, но в случае равенства сторон это условие выполняется автоматически. Если все необходимые условия выполняются, увеличиваем количество треугольников.

```
алг РавносторонниеТреугольники
нач
  цел n, k, i
  вещ x1[n], y1[n], x2[n], y2[n], x3[n], y3[n]
  вещ r1, r2, r3

  ввод n
  если n < 1 то
    вывод 'Некорректное значение'
  иначе
    для i от 1 до n
    нц
      ввод x1[i], y1[i]
      ввод x2[i], y2[i]
      ввод x3[i], y3[i]
    кц

    k = 0
    для i от 1 до n
    нц
      r1 = sqrt(sqr(x1[i] - x2[i]) + sqr(y1[i] - y2[i]))
      r2 = sqrt(sqr(x1[i] - x3[i]) + sqr(y1[i] - y3[i]))
      r3 = sqrt(sqr(x2[i] - x3[i]) + sqr(y2[i] - y3[i]))
      если r1 > 0 и r2 > 0 и r3 > 0 и r1 = r2 и r1 = r3 то
        k = k + 1
      всё
    кц
    вывод 'Количество треугольников равно ', k
  всё
кон
```

9 класс.

4. Имеется четыре стопки карточек, на каждой карточке записано два числа. Эти числа задают координаты точки на плоскости. Необходимо подсчитать, сколько перпендикулярных отрезков образуют точки, взятые с карточек попарно.

Решение. Отрезки перпендикулярны, если перпендикулярны прямые, содержащие эти отрезки. Прямые перпендикулярны, если произведение коэффициентов a в уравнениях этих прямых $y = a \cdot x + b$ равно -1. Следовательно, надо вычислить коэффициенты a_1 и a_2 для каждой прямой. Пусть (x_1, y_1), (x_2, y_2), (x_3, y_3) и (x_4, y_4) – координаты четырёх заданных точек. Тогда $a_1 = (y_1 - y_2) / (x_1 - x_2)$ и $a_2 = (y_3 - y_4) / (x_3 - x_4)$. Надо только учесть случай, когда одна из прямых параллельна оси Y – в этом случае x-координаты концов отрезка будут совпадать, и возникнет ошибка деления на 0. Поэтому для каждой четвёрки точек сначала проверяем, что $x_1 = x_2$, в этом случае если $y_3 = y_4$, то отрезки перпендикулярны. Иначе проверяем, что $x_3 = x_4$, в этом случае если $y_1 = y_2$, то отрезки перпендикулярны. Иначе вычисляем коэффициенты a_1 и a_2, и если их произведение равно -1, то отрезки перпендикулярны. В остальных случаях отрезки не перпендикулярны. Перебираем все четвёрки точек, для каждой проверяем перпендикулярность отрезков и считаем количество перпендикулярных отрезков.

```
алг ПерпендикулярныеОтрезки
нач
  цел n, k, i
  вещ x1[n], y1[n], x2[n], y2[n], x3[n], y3[n], x4[n], y4[n]
  вещ a1, a2

  ввод n
  если n < 1 то
    вывод 'Некорректное значение'
```

```
  иначе
    для i от 1 до n
    нц
      ввод x1[i], y1[i]
      ввод x2[i], y2[i]
      ввод x3[i], y3[i]
      ввод x4[i], y4[i]
    кц

    k = 0
    для i от 1 до n
    нц
      если x1[i] = x2[i] и y3[i] = y4[i] то
        k = k + 1
      иначе
        если x3[i] = x4[i] и y1[i] = y2[i] то
          k = k + 1
        иначе
          a1 = (y1 – y2) / (x1 – x2)
          a2 = (y3 – y4) / (x3 – x4)
          если a1 * a2 = -1 то
            k = k + 1
          всё
        всё
      всё
    кц
    вывод 'Количество перпендикулярных отрезков равно ', k
  всё
кон
```

9 класс.

5. Разработайте алгоритм для решения следующей задачи: найти все натуральные числа, не превосходящие заданного числа N и делящиеся нацело на квадрат каждой из своих нечётных цифр.

Решение. Переберём все числа x от 1 до N. Для каждого числа x будем последовательно вычислять его цифры и проверять, что число делится нацело на квадрат цифры, если она нечётная. Для вычисления цифр числа надо брать остаток от деления числа на 10 и после этого делить число на 10. Поскольку нам нужно проверять делимость исходного числа x, это число надо скопировать в другую переменную и именно её использовать для вычисления цифр.

```
алг НечётныеЦифры
нач
  цел n, x, y, d
  лог suit

  ввод n
  если n < 1 то
    вывод 'Некорректное значение'
  иначе
    для x от 1 до n
    нц
      y = x
      suit = истина
      повторять
        d = y mod 10
        y = y div 10
        если d mod 2 = 1 и x mod (d * d) <> 0 то
          suit = ложь
        всё
      до y = 0 или не suit
      если suit то
        вывод x
      всё
    кц
  всё
кон
```

9 класс.

6. Разработайте алгоритм для решения следующей задачи: найти все натуральные числа, не превосходящие заданного числа N и делящиеся нацело на квадрат каждой из своих цифр, находящихся на чётных позициях (позиции считаются справа налево).

Решение. Переберём все числа x от 1 до N. Для каждого числа x будем последовательно попарно вычислять его цифры и проверять, что число делится нацело на квадрат второй цифры из пары, т.е. цифры, стоящей на чётной позиции, при условии, что позиции нумеруются с 1 (если нумеровать позиции с 0, что является возможным решением, то надо брать первую цифру из пары). При этом надо проверять, что цифра не равна 0. На 1 делятся все числа, её также можно не проверять. Если в числе окажется нечётное количество цифр, то алгоритм всё равно будет работать правильно, т.к. следующая цифра вычислится как 0. Для вычисления цифр числа надо брать остаток от деления числа на 10 и после этого делить число на 10. Поскольку нам нужно проверять делимость исходного числа x, это число надо скопировать в другую переменную и именно её использовать для вычисления цифр.

```
алг ЦифрыНаЧётныхПозициях
нач
  цел n, x, y, d
  лог suit

  ввод n
  если n < 1 то
    вывод 'Некорректное значение'
  иначе
    для x от 1 до n
    нц
      y = x
      suit = истина
      повторять
        d = y mod 10
        y = y div 10
        d = y mod 10
        y = y div 10
        если d > 1 и x mod (d * d) <> 0 то
          suit = ложь
        всё
      до y = 0 или не suit
      если suit то
        вывод x
      всё
    кц
  всё
кон
```

11 класс.

7. В нотариальной конторе «Правдоруб» есть сейф. К сожалению, директор забыл код от сейфа. Но директор увлекается математикой и помнит, что код – это сумма трёх чисел Фибоначчи. Ещё он помнит, что в этом числе было не более M цифр. Числа Фибоначчи определяются как $F_{n+1} = F_n + F_{n-1}$ для $n \geq 2$, $F_0 = F_1 = 1$. Помогите директору ограничить диапазон поиска и разработайте алгоритм, который вычисляет возможные коды к сейфу, удовлетворяющие условиям выше.

Решение. Числа, состоящие не более, чем из M цифр, лежат в диапазоне от 1 до $10^M – 1$. Построим массив из чисел Фибоначчи в указанном диапазоне. Для этого первым двум элементам массива присвоим значение 1, а каждое следующее будем вычислять как сумму двух предыдущих.

Затем переберём все тройки элементов массива и будем выводить суммы, которые не превышают $10^M – 1$.

```
алг КодОтСейфа
нач
  цел fib[0..100000]
  цел m, n, i, j, k, h, s

  ввод m
  если m < 1 то
```

```
    вывод 'Некорректное значение'
  иначе

    // Вычисляем числа Фибиначчи в нужном диапазоне
    h = 10 ^ m – 1
    fib[0] = 1
    fib[1] = 1
    n = 1
    повторять
      n = n + 1
      fib[n] = fib[n – 1] + fib[n – 2]
    до fib[n] > h
    // Одно число лишнее, т.к. уже больше 10 ^ m – 1, поэтому рассматриваем элементы массива без последнего
    n = n – 1

    // Перебираем тройки чисел Фибоначчи
    для i от 1 до n – 2
    нц
      для j от i + 1 до n – 1
      нц
        для k от j + 1 до n
        нц
          s = fib[i] + fib[j] + fib[k]
          если s < h то
            вывод s
          всё
        кц
      кц
    кц
  всё
кон
```

10 класс.

8. В конторе «Рога и копыта» есть сейф. К сожалению, директор забыл код от сейфа. Но директор увлекается математикой и помнит, что код отличается от числа, свободного от квадратов, не более чем на 3. Ещё он помнит, что в этом числе было не более M цифр. Положительное число N свободно от квадратов тогда и только тогда, когда в разложении этого числа на простые множители ни одно простое число не встречается больше одного раза. Помогите директору ограничить диапазон поиска и разработайте алгоритм, который вычисляет возможные коды, удовлетворяющие условиям выше.

Решение. Числа, состоящие не более, чем из M цифр, лежат в диапазоне от 1 до $10^M - 1$. Переберём все числа N в указанном диапазоне, проверим, является ли число N свободным от квадратов, и, если это так, выведем числа $N - 3$, $N - 2$, $N - 1$, N, $N + 1$, $N + 2$, $N + 3$.

Для того, чтобы проверить, что число N свободно от квадратов, надо перебрать все возможные простые делители K числа N от 2 до $\sqrt{N}$, и проверить, делится ли N на K^2. Если N не делится на квадрат ни одного делителя, то число свободно от квадратов. Для перебора простых делителей построим массив простых чисел с помощью решета Эратосфена.

Для использования решета Эратосфена необходимо построить массив чисел в заданном диапазоне от 1 до $10^M - 1$. Поскольку число 1 не является простым, в элемент массива с индексом 1 занесём значение 0 – будет удобнее, если индекс массива и число в массиве совпадают. Затем для чисел x в диапазоне от 2 до $\sqrt{10^M - 1}$, начиная с числа x, вычёркиваем из массива (заменяем нулями) все числа с шагом x (само число x не вычёркивается). Для нахождения следующего значения x нужно найти первый незачёркнутый (ненулевой) элемент массива после текущего значения x. После этого нули можно удалить из массива.

```
цел nums[10 ^ m - 1]


алг КодОтСейфа
нач
  цел m, n

  ввод m
  если m < 1 то
    вывод 'Некорректное значение'
  иначе
```

```
    РешетоЭратосфена(10 ^ m - 1)
    для n от 1 до 10 ^ m – 1
    нц
      если СвободноОтКвадратов(n) то
        вывод n – 3, n – 2, n – 1, n, n + 1, n + 2, n + 3
      всё
    кц
  всё
кон

алг СвободноОтКвадратов(арг цел n)
нач
  цел i
  лог result

  result = истина
  i = 1
  пока nums[i] <= sqrt(n) и result
  нц
    если n mod (nums[i] * nums[i]) = 0 то
      result = ложь
    всё
    i = i + 1
  кц
  вернуть result
кон

алг РешетоЭратосфена(арг цел n)
  цел i, k

  nums[1] = 0
  для i от 2 до n
  нц
    nums[i] = i
  кц

  i = 2
  пока i <= целая_часть(sqrt(n))
  нц
    для j от 2 * i до n шаг i
    нц
      nums[j] = 0
    кц
    выполнить
      i = i + 1
    до nums[i] <> 0
  кц
  k = 1
  для i от 1 до n
  нц
    если nums[i] <> 0 то
      nums[k] = nums[i]
      k = k + 1
    всё
  кц
кон
```

10 класс.

9. Магический, или волшебный квадрат – это квадратная таблица $n \times n$, заполненная n^2 различными числами таким образом, что сумма чисел в каждой строке, каждом столбце и на обеих диагоналях одинакова. Магический квадрат называют нормальным, если он заполнен натуральными числами от 1 до n^2. Рамочный квадрат – это такой магический квадрат, что если в нем отбросить окаймляющие «полосы» шириной в одну или нескольких клеток, то оставшийся квадрат не утратит своего магического свойства. Разработайте алгоритм проверки, является ли заданная таблица нормальным магическим квадратом, и определите сколько полос можно отбросить, чтобы оставшаяся часть также была нормальным магическим квадратом.

Решение. Для проверки того, что квадрат является нормальным магическим квадратом, необходимо сначала проверить, какими числами заполнена таблица. Будем искать в ней числа от 1 до n^2. Если какое-то число не найдётся, значит, таблица не является нормальным магическим квадратом.

После этого надо подсчитать сумму элементов первой строки и сравнить её с суммами элементов других строк и суммами элементов столбцов, а также с суммами элементов на диагоналях. Если какие-то суммы не равны, то квадрат не является магическим.

Для определения количества полос, которые можно откинуть, нужно реализовать проверку того, что квадрат является магическим в виде функции, и передавать в эту функцию либо размер таблицы и количество отбрасываемых полос, либо номера рассматриваемых строк и столбцов. С помощью этой функции можно будет проверить остаётся ли квадрат магическим, если отбросить 0, 1, 2, …, (n – 1) / 2 полосы.

```
алг МагическийКвадрат
нач
  цел n
  цел table[n, n]
  цел i, j

  ввод n
  если n < 1 то
    вывод 'Некорректное значение'
  иначе
    для i от 1 до n
    нц
      для j от 1 до n
      нц
        ввод table[i, j]
      кц
    кц
    если НормальныйМагическийКвадрат(table, 1, n, 1, n) то
      вывод 'Таблица является магическим квадратом'
      для i от 1 до (n – 1) div 2
      нц
        если НормальныйМагическийКвадрат(table, 1 + i, n – i, 1 + i, n – i) то
          вывод 'Таблица является рамочным магическим квадратом. Ширина отбрасываемой полосы ', i
        всё
      кц
    иначе
      вывод 'Таблица не является магическим квадратом'
    всё
  всё
кон

алг НормальныйМагическийКвадрат(арг цел table[n, n], арг цел rs, re, cs, ce)
нач
  цел i, j, k, sum, s
  лог exist

  // Проверяем наличие чисел от 1 до (размер квадрата) ^ 2
  exist = истина
  k = 1
  пока k <= sqr(re – rs + 1) и exist
  нц
    exist = ложь
    i = rs
    пока i <= re и не exist
    нц
      j = cs
      пока j <= cr и не exist
      нц
        если table[i, j] = k то
          exist = истина
        всё
        j = j + 1
      кц
      i = i + 1
    кц
    k = k + 1
  кц

  если не exist то
    вернуть ложь
  всё

  // Считаем сумму первой строки
  sum = 0
  для j от cs до ce
```

```
  нц
    sum = sum + table[rs, j]
  кц

  // Проверяем, что остальные строки имеют такую же сумму
  для i от rs + 1 до re
  нц
    s = 0
    для j от cs до ce
    нц
      s = s + table[i, j]
    кц
    если s <> sum то
      вернуть ложь
    всё
  кц

  // Проверяем, что столбцы имеют такую же сумму
  для j от cs до ce
  нц
    s = 0
    для i от rs до re
    нц
      s = s + table[i, j]
    кц
    если s <> sum то
      вернуть ложь
    всё
  кц

  // Проверяем главную диагональ
  s = 0
  для i от rs до re
  нц
    s = s + table[i, i]
  кц
  если s <> sum то
    вернуть ложь
  всё

  // Проверяем побочную диагональ
  s = 0
  для i от rs до re
  нц
    s = s + table[i, re – i + rs]
  кц
  если s <> sum то
    вернуть ложь
  всё

  вернуть истина
кон
```

11 класс.

10. Подчитайте сумму ряда, представленного ниже, наиболее эффективным способом. Вычисления остановить, когда будет достигнута точность ε, т.е. модуль очередного слагаемого станет меньше точности. Известно, что $|x| \leq 1$.

$$1-x^2\left(\tfrac{1}{2!}-\tfrac{1}{1}\right)+x^4\left(\tfrac{1}{4!}-\tfrac{1}{2}\right)-\ldots\pm x^{2i}\left(\tfrac{1}{(2i)!}-\tfrac{1}{i}\right)\mathrm{m}\ldots$$

Решение. Для вычисления суммы ряда наиболее эффективным способом необходимо вычислять каждого слагаемое, умножая предыдущее слагаемое на рекуррентное соотношение. Но в данном случае приводить выражения в скобках к общему знаменателю сложно и бесполезно, т.к. суммы в числителе сокращаться не будут. Но можно разложить ряд на три составляющие: x^{2i}, $\frac{1}{(2i)!}$ и $\frac{1}{i}$. Также в первой составляющей учтём перемену знака слагаемых. Тогда для первой составляющей рекуррентное соотношение есть $-x^2$, для второй $\frac{1}{(2i-1)(2i)}$, третья составляющая вычисляется без использования рекуррентного соотношения. Кроме того, собственно ряд начинается с выражения $-x^2\left(\frac{1}{2!}-\frac{1}{1}\right)$, единицу надо прибавить дополнительно.

Положим сумму ряда s равной 1, переменную a_1 равной $-x^2$, переменную a_2 равной 0.5, номер слагаемого i равным 1, слагаемое a вычисляется как $a_1 \cdot (a_2 - 1 / i)$. Пока модуль слагаемого больше или равен точности, прибавляем слагаемое к общей сумме, увеличиваем i на 1, умножаем переменные a_1 и a_2 на соответствующее рекуррентное соотношение и вычисляем следующее слагаемое.

```
алг СуммаРяда
нач
  вещ x, e, a, a1, a2, s
  цел i

  ввод x, e
  если abs(x) > 1 то
    вывод 'Неверное значение аргумента'
  иначе
    если e <= 0 или e >= 1 то
      вывод 'Неверное значение точности'
    иначе
      s = 1
      a1 = -x * x
      a2 = 0.5
      i  = 1
      a  = a1 * (a2 – 1 / i)
      пока abs(a) >= e
      нц
        s  = s + a
        i  = i + 1
        a1 = a1 * -x * x
        a2 = a2 / (2 * i – 1) / (2 * i)
        a  = a1 * (a2 – 1 / i)
      кц
    всё
  всё
кон
```

10 класс.

11. В прямоугольной таблице размером $M \times N$ записаны целые числа. Определить номер строки с максимальным количеством элементов, не кратных заданному числу (считать, что она единственная).

Решение. Подсчитаем количество элементов, не кратных заданному числу, в первой строке и присвоим это значение переменной *max*. Также присвоим значение 1 переменной *imax* – номер искомой строки. Далее будем считать количество элементов, не кратных заданному числу, во 2-ой, 3-ей, …, M-ой строке и сравнивать это значение с переменной *max*. Если в текущей строке количество элементов, не кратных заданному числу, больше максимума, присвоим переменным *max* и *imax* новые значения.

```
алг МаксимумНекратных
нач
  цел m, n, p
  цел table[m, n]
  цел max, imax, i, j, k

  ввод m
  если m < 1 или n < 1 или p < 1 то
    вывод 'Некорректные входные данные'
  иначе
    для i от 1 до m
    нц
      для j от 1 до n
      нц
        ввод table[i, j]
      кц
    кц
    k = 0
    для j от 1 до n
    нц
      если table[1, j] mod p <> 0 то
```

```
        k = k + 1
      всё
    кц
    max  = k
    imax = 1
    для i от 2 до m
    нц
      k = 0
      для j от 1 до n
      нц
        если table[i, j] mod p <> 0 то
          k = k + 1
        всё
      кц
      если k > max то
        max  = k
        imax = 1
      всё
    кц
    вывод 'Номер строк с максимальным количеством элементов, не кратных ', p, ', есть ', imax
  всё
кон
```