

ЗАДАНИЕ ПО ИНФОРМАТИКЕ
ВАРИАНТ 32101 для 10 класса

Для заданий 1-5 требуется разработать алгоритмы на языке блок-схем, псевдокоде или естественном языке.

1. Найти сумму ряда чисел с точностью ε , $|x| \leq 1$
- $$-1 + \frac{1}{\sqrt[4]{1-x}} = \frac{1x}{4} + \frac{1 \cdot 5x^2}{4 \cdot 8} + \frac{1 \cdot 5 \cdot 9x^3}{4 \cdot 8 \cdot 12} + \frac{1 \cdot 5 \cdot 9 \cdot 13x^4}{4 \cdot 8 \cdot 12 \cdot 16} + \dots + \frac{1 \cdot 5 \cdot 9 \cdot 13 \cdot \dots \cdot (4n-3)x^n}{4 \cdot 8 \cdot 12 \cdot 16 \cdot \dots \cdot (4n)}$$

Решение. Положим переменную a равной первому слагаемому $\frac{1x}{4}$, а переменную s равной переменной a . Далее в цикле переменную a надо умножать на рекуррентное соотношение, которое в данном случае равно $\frac{(4n-3)x}{(4n)}$, и прибавлять к переменной s . Цикл продолжается пока модуль очередного слагаемого не станет меньше точности ε .

```
алг СуммаРяда()
нач
  вещ x, e, a, s
  цел n

  ввод x, e
  если abs(x) > 1 или e > 1 или e <= 0 то
    вывод "Некорректные исходные данные"
  иначе

    a = x / 4
    s = a
    n = 2
    пока abs(a) >= e
      нц
        a = a * (4 * n - 3) * x / (4 * n)
        s = s + a
        n = n + 1
      кц
    вывод s

  всё
кон
```

2. Автоморфным называют число, десятичная запись квадрата которого оканчивается цифрами самого этого числа. Найти произведение чётных автоморфных чисел с порядковыми номерами от P до Q . Среди таких чисел найти все дружественные числа. Два числа называются дружественными, если каждое из них равно сумме всех делителей другого, кроме самого этого числа.

Решение. Перебираем натуральные числа, пока не найдём автоморфное число с номером $P-1$. Далее перебираем следующие натуральные числа, пока не найдём автоморфное число с номером Q . При этом запоминаем чётные автоморфные числа, а также считаем их произведение. Далее надо найти дружественные числа. Для каждого числа x из массива найденных чётных автоморфных чисел подсчитаем сумму его делителей. Для этого нужно проверить, на какие числа из диапазона от 2 до $x/2$ делится число x , и сложить эти числа, а также прибавить 1 (т.к. любое число делится на 1). Далее перебираем все пары чисел x и y из сформированного массива, и если x равно сумме делителей y , а y равно сумме делителей x , то выводим числа x и y .

Для проверки на автоморфность надо возвести число в квадрат и сравнить последние цифры нового числа с исходным. Для этого надо взять остаток от деления нового числа на 10 в некоторой степени. Степень зависит от количества цифр в исходном числе. Поэтому исходное число делим несколько раз на 10, пока не получится 0,

Олимпиада школьников «Надежда энергетики». Отборочный этап. Очная форма.

одновременно столько же раз умножаем на 10 делитель, изначально равный 1. Затем берём остаток от деления нового числа на полученный делитель, и если этот остаток от деления равен исходному числу, то исходное число является автоморфным.

Для того чтобы не вычислять делитель для каждого числа можно поступить следующим образом. Сначала положим делитель равным 10. Когда проверяемое число станет равным делителю, умножаем делитель на 10.

алг АвтоморфныеЧисла()

нач

цел p, q, n, i, j, k, d, amn[q], sum[q]

ввод x, e

если p <= 0 или q <= 0 или u <= 0 или p > q то

вывод "Некорректные исходные данные"

иначе

n = 0

i = 0

d = 10

пока i <= p - 1

нц

если n ^ 3 mod d = n то

i = i + 1

всё

n = n + 1

если n = d то

d = d * 10

всё

кц

k = 0

пока i <= q

нц

если n ^ 3 mod d = n то

i = i + 1

если n mod 2 = 0 то

k = k + 1

amn[k] = n

sum[k] = СуммаДелителей(n)

всё

всё

n = n + 1

если n = d то

d = d * 10

всё

кц

для i от 1 до k - 1

нц

для j от i + 1 до k

нц

если sum[i] = sum[j] то

вывод amn[i], amn[j]

всё

кц

кц

всё

кон

алг СуммаДелителей(арг цел n)

нач

цел s, i

s = 1

для i от 2 до n div 2

нц

если n mod i = 0 то

s = s + i

всё

кц

вернуть s

кон

3. На координатной плоскости по линиям сетки построено несколько прямоугольников. Необходимо подсчитать число точек с целочисленными координатами, принадлежащими сразу всем этим прямоугольникам.

Решение. Будем считать, что прямоугольник задаётся четырьмя числами – координаты x и y левого нижнего угла и координаты x и y правого верхнего угла. Числа должны быть целыми. Можно использовать четыре одномерных массива. Найдём прямоугольник, являющийся пересечением всех исходных прямоугольников. Для этого положим искомым прямоугольником равным первому, и будем последовательно искать пересечение искомого прямоугольника со 2-ым, 3-им и т.д. Для нахождения пересечения двух прямоугольников надо сделать следующее:

1. запоминаем длины сторон прямоугольников;
2. из левых координат двух прямоугольников выбираем минимальную и записываем в переменную x_{min} , а другую координату сохраняем как левую координату прямоугольника, являющегося пересечением;
3. из правых координат двух прямоугольников выбираем максимальную и записываем в переменную x_{max} , а другую координату сохраняем как правую координату прямоугольника, являющегося пересечением;
4. если $x_{max} - x_{min}$ больше, чем сумма длин сторон по оси x , значит, прямоугольники не пересекаются;
5. из нижних координат двух прямоугольников выбираем минимальную и записываем в переменную y_{min} , а другую координату сохраняем как нижнюю координату прямоугольника, являющегося пересечением;
6. из верхних координат двух прямоугольников выбираем максимальную и записываем в переменную y_{max} , а другую координату сохраняем как верхнюю координату прямоугольника, являющегося пересечением;
7. если $y_{max} - y_{min}$ больше, чем сумма длин сторон по оси y , значит, прямоугольники не пересекаются.

Если прямоугольники не пересекаются, можно прекратить перебор прямоугольников. После нахождения пересечения всех прямоугольников количество точек с целочисленными координатами вычисляется через координаты прямоугольника, являющегося пересечением, по формуле $(x_{прав} - x_{лев} + 1) \cdot (y_{ниж} - y_{верх} + 1)$.

алг ПересечениеПрямоугольников()

нач

цел $n, x1[n], xp[n], yn[n], yv[n], rx1, rxp, ryn, ryv, xmin, xmax, ymin, ymax, a, b, ar, br, i$
лог intersec

ввод n

если $n \leq 0$ то

 вывод "Некорректные исходные данные"

иначе

 для i от 1 до n

 нц

 ввод $x1[i], xp[i], yn[i], yv[i]$

 кц

$rx1 = x1[1]$

$rxp = xp[1]$

$ryn = yn[1]$

$ryv = yv[1]$

$i = 2$

 intersec = истина

 пока $i \leq n$ и intersec

 нц

$a = xp[i] - x1[i]$

$b = yv[i] - yn[i]$

$ar = rxp - rx1$

Олимпиада школьников «Надежда энергетики». Отборочный этап. Очная форма.

```
br = ryv - ryn
если xl[i] < rxl то
  xmin = xl[i]
иначе
  xmin = rxl
  rxl = xl[i]
всё
если xp[i] > rxp то
  xmax = xp[i]
иначе
  xmax = rxp
  rxp = xp[i]
всё
если xmax - xmin > a + ra то
  intersec = ложь
всё
если yn[i] < ryn то
  ymin = yn[i]
иначе
  ymin = ryn
  ryn = yn[i]
всё
если yv[i] > ryv то
  ymax = yv[i]
иначе
  ymax = ryv
  ryv = yv[i]
всё
если ymax - ymin > b + rb то
  intersec = ложь
всё
i = i + 1
кц

если intersec то
  вывод "Количество точек = ", (rxp - rxl + 1) * (ryv - ryn + 1)
иначе
  вывод "Прямоугольники не пересекаются"
всё

всё
кон
```

4. Рассмотрим возрастающий ряд всех положительных несократимых правильных дробей, знаменатель которых меньше или равен n . Разработайте алгоритм нахождения суммы P тех членов данного ряда, для которых знаменатель является совершенным числом. Число называется совершенным, если оно равно сумме всех своих делителей, исключая само это число.

Решение. Перебираем все числа x в диапазоне от 2 до n . Для каждого числа x надо проверить, является ли оно совершенным. Для этого рассмотрим возможные делители i в диапазоне от 2 до $x/2$ (любое число делится на 1, поэтому изначально сумма делителей должна быть равна 1, а последний возможный делитель числа x , не равный x , есть $x/2$), и если x делится на i , то прибавляем i к сумме делителей. Если сумма делителей равна самому числу, значит, число является совершенным. В этом случае перебираем все возможные числители дроби, которые находятся в пределах от 1 до $x - 1$. Если дробь является несократимой, запоминаем её. Дробь является несократимой, если НОД числителя и знаменателя равен 1. Для нахождения НОД используем алгоритм Евклида: из большего числа надо вычитать меньшее, пока числа не станут равными.

Для ускорения работы алгоритма можно использовать не простой, а расширенный алгоритм Евклида, который заключается в следующем. Сначала надо составить два исходных уравнения.

$$x \cdot u_1 + y \cdot v_1 = x$$

$$x \cdot u_2 + y \cdot v_2 = y$$

Для того чтобы уравнения выполнялись, коэффициенты должны иметь следующие значения: $u_1 = 1$, $v_1 = 0$, $u_2 = 0$, $v_2 = 1$. После этого из первого уравнения вычитается второе, умноженное на результат целочисленного деления x на y (обозначим как q).

$$x \cdot (u_1 - q \cdot u_2) + y \cdot (v_1 - q \cdot v_2) = x - q \cdot y$$

В правой части уравнения получается остаток от деления x на y . На следующем шаге те же действия выполняются над вторым и третьим уравнениями. Алгоритм прекращает работу, когда правая часть уравнения становится равной 0. Значение в правой части уравнения на предпоследнем шаге даёт наибольший общий делитель чисел x и y .

Чтобы сложить запомненные дроби, надо найти НОК всех знаменателей, привести дроби к общему знаменателю, сложить числители и упростить дробь.

НОК нескольких чисел вычисляется через последовательные вычисления НОК пар чисел – $\text{НОК}(\text{НОК}(a_1, a_2, \dots, a_{n-1}), a_n)$. Получить НОК двух чисел можно,

воспользовавшись формулой $\text{НОК}(x, y) = \frac{x \cdot y}{\text{НОД}(x, y)}$.

алг СуммаДробей()

нач

цел n, x, y, num[n * n], den[n * n], k, nok, p, wr, i

ввод n

если n <= 0 то

 вывод "Некорректные исходные данные"

иначе

 k = 0

 для x от 2 до n

 нц

 если x = СуммаДелителей(x) то

 для y от 1 до x - 1

 нц

 если НОД(x, y) = 1 то

 k = k + 1

 num[k] = y

 den[k] = x

 всё

 кц

 всё

 кц

 nok = den[1]

 для i от 2 до k

 нц

 nok = nok * den[i] / НОД(nok, den[i])

 кц

 p = 0

 для i от 1 до k

 нц

 p = p + num[i] * nok / den[i]

 кц

 wr = p div nok

 p = p mod nok

 вывод "Результат = "

 если wr > 0 то

 вывод wr, " "

 всё

 вывод p, " / ", nok

всё

кон

алг СуммаДелителей(арг цел n)

нач

```

цел s, i

s = 1
для i от 2 до n div 2
нц
    если n mod i = 0 то
        s = s + i
    всё
кц
вернуть s
кон

алг НОД(арг цел x, y)
нач
    цел q, r, u1, u2, v1, v2, u, v, d

    u1 = 1
    u2 = 0
    v1 = 0
    v2 = 1

    пока y > 0
    нц
        q = x div y
        r = x mod y

        u = u1 - q * u2
        v = v1 - q * v2

        x = y
        y = r

        u1 = u2
        u2 = u
        v1 = v2
        v2 = v
    кц

    вернуть x
кон
    
```

5. Сопутствующие числа Пелля задаются соотношением:

$$P_n = \begin{cases} 2, n = 0 \\ 2, n = 1 \\ 2P_{n-1} + P_{n-2}, n > 1 \end{cases} . \text{ Разработайте алгоритм, находящий сопутствующие числа}$$

Пелля в диапазоне от P до Q , которые являются простыми.

Решение. Создадим массив и проинициализируем его первые два элемента значением 2. Далее будем вычислять следующие элементы массива – каждый n -ый элемент вычисляется по формуле $2P_{n-1} + P_{n-2}$. Прекращаем вычисления, когда очередной элемент массива будет больше или равен Q . Далее надо в полученном массиве найти простые числа.

Можно обойтись без массива. Объявляем две переменных и инициализируем их значением 2. Далее вычисляем значение третьей переменной по такой же формуле. Пока значение третьей переменной меньше P , просто вычисляем эти значения. Далее пока получаемые значения меньше или равны Q , проверяем их на простоту. При переходе к следующему шагу цикла надо сдвинуть значения, записав вторую переменную в первую, а третью – во вторую.

Все найденные числа надо проверить на простоту. Можно примитивно проверять делится ли число x на какое-либо другое число из диапазона от 2 до \sqrt{x} . А можно построить массив простых чисел в диапазоне от 2 до Q с помощью решета Эратосфена и проверять каждое число Пелля на попадание в этот массив.

Олимпиада школьников «Надежда энергетики». Отборочный этап. Очная форма.

Для использования решета Эратосфена необходимо построить массив чисел в заданном диапазоне от 1 до Q . 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 i в диапазоне от 2 до \sqrt{Q} , начиная с числа i , вычёркиваем из массива (заменяем нулями) все числа с шагом i (само число i не вычёркивается). Для нахождения следующего значения i нужно найти первый незачёркнутый (ненулевой) элемент массива после текущего значения i .

цел nums[q]

алг ПростыеЧислаПелля()

нач

цел p, q, p1, p2, p3

ввод p, q

если p <= 0 или q <= 0 или p > q то

вывод "Некорректные исходные данные"

иначе

РешетоЭратосфена(q)

p1 = 2

p2 = 2

p3 = 2 * p2 + p1

пока p3 < p

нц

p1 = p2

p2 = p3

p3 = 2 * p2 + p1

кц

пока p3 <= q

нц

если nums[p3] <> 0 то

вывод p3

всё

p1 = p2

p2 = p3

p3 = 2 * p2 + p1

кц

всё

кон

алг РешетоЭратосфена(арг цел n)

цел i

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

кон