ЗАДАНИЕ ПО ИНФОРМАТИКЕ
ВАРИАНТ 31991 для 9 класса

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

1. В теории чисел нечётное натуральное число k называют числом Серпинского, если для любого натурального числа n число $k \times 2^n + 1$ является составным. Разработайте алгоритм поиска чисел Серпинского для k = U в диапазоне для n от P до Q.

Решение. Необходимо перебрать все числа n в диапазоне от P до Q и для каждого n проверить, что число $s = k \times 2^n + 1$ является составным. Для этого необходимо проверить, делится ли число s на какое-то другое число в диапазоне от 3 до $\sqrt{s}$ (на 2 не делится, т.к. 2^n – чётное число, следовательно, $k \times 2^n + 1$ – нечётное число). Для упрощения проверки можно заранее построить массив простых чисел в диапазоне от 2 до Q с помощью решета Эратосфена.

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[1000000]

алг ЧислаСерпинского()
нач
  цел p, q, k, u, n, s

  ввод p, q, u
  если p <= 0 или q <= 0 или u <= 0 или p > q то
    вывод "Некорректные исходные данные"
  иначе

    k = u
    s = k * 2 ^ q + 1

    РешетоЭратосфена(s)

    для n от p до q
    нц
      s = k * 2 ^ n + 1
      если nums[s] = 0 то
        вывод s
      всё
    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
```

2. Известно, что десятизначное число A = 2013x2013y делится нацело на 121. Составьте алгоритм для нахождения всех возможных пар цифр (x, y).

Решение – 1 способ. Перебираем все x в диапазоне от 0 до 9 и для каждого x перебираем все y в диапазоне от 0 до 9. Для всех возможных пар проверяем остаток от деления на 121.

Решение – 2 способ. Чтобы число делилось на 121, оно должно дважды делится на 11. На 11 число делится, если разность между суммой цифр в чётных позициях и суммой цифр в нечётных позициях делится на 11. Для числа 2013x2013y эта разность равна (2 + 1 + x + 0 + 3) – (0 + 3 + 2 + 1 + y) = x – y. Поскольку x и y – цифры, x – y может делится на 11, только если x = y. Поэтому можно перебрать все x в диапазоне от 0 до 9, для каждого x вычисляем результат деления числа 2013x2013x на 11 и проверить, делится ли этот результат на 11.

Решение – 3 способ. Чтобы число делилось на 121, оно должно дважды делится на 11. На 11 число делится, если разность между суммой цифр в чётных позициях и суммой цифр в нечётных позициях делится на 11. Для числа 2013x2013y эта разность равна (2 + 1 + x + 0 + 3) – (0 + 3 + 2 + 1 + y) = x – y. Поскольку x и y – цифры, x – y может делится на 11, только если x = y. Представим число 2013x2013x в следующем виде $2013 \cdot 10^6 + x \cdot 10^5 + 2013 \cdot 10 + x$. 2013 / 11 = 183, значит, $2013 \cdot 10^6 + 2013 \cdot 10 \ / \ 11 = 183 \cdot 10^6 + 183 \cdot 10$. Поскольку в этом числе одинаковые цифры оказываются в разных (чётных и нечётных) позициях, разность между их суммами равно 0, и число делится на 11. Значит, $2013 \cdot 10^6 + 2013 \cdot 10$ делится на 121. Следовательно, необходимо чтобы число $x \cdot (10^5 + 1)$ тоже делилось на 121. $10^5 + 1$ делится на 11, результат равен 9091. Но это число не делится на 11. Поэтому $x \cdot (10^5 + 1)$ может делиться на 121 только при $x = 0$.

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

Решение. Перебираем все числа i в диапазоне от 1 до N. Для каждого числа i надо найти его цифры (для чего последовательно нужно делить его на 10) и проверить остаток от деления i на куб каждой из его цифр.

```
алг ДелениеНаКубЦифр()
нач
  цел n, i, k, d
  лог f

  ввод n

  если n <= 0 то
    вывод "Некорректные исходные данные"
  иначе

    для i от 1 до n
    нц
      k = i
      f = истина
      пока k > 0 и f
```

```
      нц
        d = k mod 10
        k = k div 10
        если i mod d <> 0 то
          f = ложь
        всё
      кц
      если f то
        вывод i
      всё
    кц
  всё
кон
```

4. Даны три стопки карточек, на каждой из которых записано два числа. Эти два числа задают координаты точки на плоскости. Посчитать число треугольников общего вида и прямоугольных треугольников среди троек.

Решение. Для каждой тройки точек (x_1, y_1), (x_2, y_2) и (x_3, y_3) надо проверить, можно ли построить треугольник. Для этого надо найти три расстояния между парами точек по формуле $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ и проверить, что сумма каждых двух расстояний больше третьего. Чтобы треугольник был прямоугольным, скалярное произведение каких-то двух векторов, образованных сторонами треугольника, должно быть равно 0, т.е., например, $(x_3 - x_1) \cdot (x_2 - x_1) + (y_3 - y_1) \cdot (y_2 - y_1)$ должно быть равно 0.

```
алг Треугольники()
нач
  цел n, x1[n], y1[n], x2[n], y2[n], x3[n], y3[n], r1, r2, r3, i, t, rt
  лог f

  ввод n

  если n <= 0 то
    вывод "Некорректные исходные данные"
  иначе

    t = 0
    rt = 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 + r2 > r3 и r1 + r3 > r2 и r2 + r3 > r1 то
        t = t + 1
        если (x3[i] - x1[i]) * (x2[i] - x1[i]) + (y3[i] - y1[i]) * (y2[i] - y1[i]) = 0 или
             (x1[i] - x2[i]) * (x3[i] - x2[i]) + (y1[i] - y2[i]) * (y3[i] - y2[i]) = 0 или
             (x1[i] - x3[i]) * (x2[i] - x3[i]) + (y1[i] - y3[i]) * (y2[i] - y3[i]) = 0 то
          rt = rt + 1
        всё
      всё
    кц
    вывод "Количество треугольников общего вида   - ", t
    вывод "Количество прямоугольных треугольников - ", rt

  всё
кон
```

5. Найти количество пар взаимно простых чисел из диапазона от P до Q, в которых хотя бы одно число из пары является совершенным.

Решение. Перебираем все возможные пары чисел из диапазона от P до Q. Для каждой пары проверяем, что числа являются взаимно простыми, т.е. их наибольший общий делитель равен 1. Если это так, то надо проверить, что хотя бы одно число является совершенным. Для проверки того, что число n является совершенным, надо последовательно проверять числа в диапазоне от 1 до n / 2 и суммировать те, которые являются делителями числа n. Если сумма делителей равна n, значит, число n является совершенным.

Для поиска НОД можно использовать простой или расширенный алгоритм Евклида. Простой алгоритм Евклида заключается в вычитании меньшего числа из большего, пока числа не станут равными. Расширенный алгоритм 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.

```
алг ПарыЧисел()
нач
  цел p, q, m, n, k

  ввод p, q, u
  если p <= 0 или q <= 0 или u <= 0 или p > q то
    вывод "Некорректные исходные данные"
  иначе

    k = 0
    для m от p до q – 1
    нц
      для n от m + 1 до q
      нц
        если НОД(m, n) = 1 и (СовершенноеЧисло(n) или СовершенноеЧисло(m)) то
          k = k + 1
        всё
      кц
    кц

  всё
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

алг СовершенноеЧисло(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
  если s = n то
    вернуть истина
  иначе
    вернуть ложь
  всё
кон
```