ЗАДАНИЕ ПО ИНФОРМАТИКЕ
ВАРИАНТ 73111 для 11 класса

Для заданий 1, 2, 4, 5 требуется разработать алгоритм на языке блок-схем, псевдокоде или естественном языке

1. Дана последовательность чисел C_1, C_2, ..., C_n, ... в которой C_n есть последняя цифра числа n^n. Первые 4 элемента последовательности таковы: 1 ($1^1 = 1$), 4 ($2^2 = 4$), 7 ($3^3 = 27$), 6 ($4^4 = 256$). Разработать алгоритм нахождения наименьшего периода этой последовательности. Предусмотреть выход из алгоритма, если возможная величина периода превысила 10! (факториал числа n определяется как произведение всех натуральных чисел от 1 до n).

Решение. Первая подзадача – вычислить последнюю цифру числа n^n. Поскольку число такого вида может оказаться очень большим, лучше вычислить именно последнюю цифру, т.е. остаток от деления на 10. Для этого можно положить переменную *result* равной n **mod** 10, затем (n – 1) раз произвести следующие вычисления:

$result = (result \cdot n)$ **mod** 10

Далее будем искать период последовательности. Если период равен p, то для проверки этого нужно вычислить $2 \cdot p$ элементов последовательности. Сначала вычислим первые два элемента последовательности. Затем для p от 1 до 10! проверяем, является ли p периодом последовательности. Если да, то прекращаем вычисления, в противном случае добавляем к последовательности ещё два элемента и проверяем следующее значение p.

Для проверки, что некоторое значение p является периодом последовательности длиной $2 \cdot p$, надо проверить, что для всех i от 1 до p элемент последовательности с номером i равен элементу последовательности с номером $i + p$.

```
алг Период
нач
  цел n, p, c[10000000]
  цел factorial10 = 3628800
  лог period

  p = 0
  n = 1
  period = ложь
  пока p < factorial10 и не period
  нц
    p = p + 1
    c[n] = ПоследняяЦифра(n)
    c[n + 1] = ПоследняяЦифра(n + 1)
    n = n + 2
    period = истина
    i = 1
    пока i <= p и period
    нц
      если c[i] <> c[i + p] то
        period = ложь
      всё
      i = i + 1
    кц
  кц
  если period то
    вывод p
  иначе
    вывод 'Период не найден'
  всё
кон
```

```
алг ПоследняяЦифра(арг цел n)
нач
  цел result, i

  result = n mod 10
  для i от 1 до n – 1
  нц
    result = (result * n) mod 10
  кц
  вернуть result
кон
```

2. Дима и Петя играли в игру. На столе есть 3 пересекающихся ряда карточек, на каждой из которых записано целое число. Ряды расположены в виде треугольника, крайние карточки в каждом ряду представляют углы треугольника. Помогите ребятам и разработайте алгоритм для решения следующей задачи. Задача – расположить все карточки в первом ряду от меньшего числа к большему, во втором ряду – от большего к меньшему. Значения на стыках рядов выставлять по порядку обхода. В третьем ряду числа должны располагаться от минимального к максимальному. Если на стыках с двумя другими рядами не оказался минимум и максимум, вывести соответствующее сообщение.

Решение. Для хранения данных можно использовать три одномерных массива, но тогда придётся следить за тем, чтобы первый/последний элемент в одном ряду был равен первому/последнему элементу в других рядах. Или можно использовать один одномерный массив, в котором элементы с номерами от 1 до n будут хранить числа первого ряда, элементы с номерами от n до $2 \cdot n - 1$ будут хранить числа второго ряда, и, наконец, элементы с номерами от $2 \cdot n - 1$ до $3 \cdot n - 2$ будут хранить числа третьего ряда. В элемент с номером $3 \cdot n - 1$ можно после сортировки первого ряда скопировать значение первого элемента для упрощения обработки третьего ряда.

Чтобы использовать одну процедуру для сортировки всех рядов надо добавить в параметры флаг для управления направлением сортировки (по возрастанию или по убыванию), а также номера первого и последнего элемента для обработки (если данные хранятся в одном массиве). Используем любой метод сортировки, при этом элементы первого ряда сортируем по возрастанию, затем элементы второго ряда – по убыванию. Далее проверяем, что первый элемент третьего ряда меньше всех остальных элементов третьего ряда, а последний элемент третьего ряда – больше всех остальных элементов третьего ряда. Если это так, сортируем третий ряд по возрастанию. В противном случае проверяем, что первый элемент третьего ряда больше всех остальных элементов третьего ряда, а последний элемент третьего ряда – меньше всех остальных элементов третьего ряда. В этом случае сортируем третий ряд по убыванию. В противном случае выводим сообщение, что сортировка невозможна.

```
алг Ряды
нач
  цел row[1000]                          // Будем использовать один массив для хранения всех рядов
  цел n, i
  лог min1, maxN, max1, minN


  ввод n
  если n <= 0 то
    вывод 'Неверное количество элементов'
  иначе
    для i от 1 до n
    нц
      ввод row[i]
    кц
    для i от n + 1 до 2 * n - 1
    нц
```

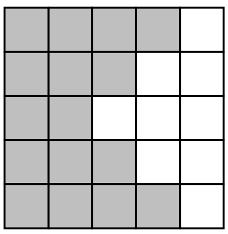
```
      ввод row[i]
    кц
    для i от 2 * n до 3 * n - 2
    нц
      ввод row[i]
    кц

    QuickSort(row, 1, n, истина)
    QuickSort(row, n, 2 * n - 1, ложь)

    min1 = истина
    maxN = истина
    max1 = истина
    minN = истина
    для i от 2 * n до 3 * n - 2
    нц
      если row[i] < row[1] то
        min1 = ложь
      всё
      если row[i] > row[2 * n – 1] то
        maxN = ложь
      всё
      если row[i] > row[1] то
        max1 = ложь
      всё
      если row[i] < row[2 * n – 1] то
        minN = ложь
      всё
    кц

    если min1 и maxN то
      QuickSort(row, 2 * n, 3 * n - 2, ложь)
    иначе
      если max1 и minN
        QuickSort(row, 2 * n, 3 * n - 2, истина)
      иначе
        вывод 'Невозможно отсортировать третий ряд'
      всё
    всё

    для i от 1 до n
    нц
      вывод row[i]
    кц
    для i от n + 1 до 2 * n - 1
    нц
      вывод row[i]
    кц
    для i от 2 * n до 3 * n - 2
    нц
      вывод row[i]
    кц
  всё
кон

алг QuickSort(арг рез цел x[10000], арг цел n1, n2, арг лог ascending)
нач
  цел i, j,
  вещ y, k

  если n2 - n1 = 1 то
    если (не ascending и x[n1] < x[n2]) или (ascending и x[n1] > x[n2]) то
      y = x[n1]
      x[n1] = x[n2]
      x[n2] = y
    всё
  иначе
    если n2 - n1 > 1 то
      k = x[(n1 + n2) div 2]
      i = n1
      j = n2
      повторять
        пока (не ascending и x[i] > k) или (ascending и x[i] < k)
        нц
```

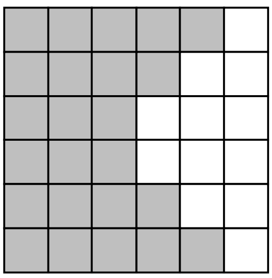
```
        i = i + 1
      кц
      пока (не ascending и x[j] < k) или (ascending и x[j] > k)
      нц
        j = j - 1
      кц
      если i <= j то
        y = x[i]
        x[i] = x[j]
        x[j] = y
        i = i + 1
        j = j - 1
      всё
    до i > j
    QuickSort(x, n1, j, ascending)
    QuickSort(x, i, n2, ascending)
  всё
 всё
кон
```

3. Участник тематической смены «Школа молодого энергетика» во Всероссийском детском центре «Смена» на берегу Черного моря Серёжа всегда любил играть с калькулятором и носил его с собой. В перерыве между занятиями Сережа решил поделить два вещественных числа a и b друг на друга, а затем результат снова разделил на b. Выполнив эти действия много раз (много делений на b), Серёжа неожиданно для себя увидел на дисплее калькулятора ноль. Сережа - очень интересующийся школьник, и попробовал повторить результаты на домашнем компьютере. В итоге Серёжа снова получил ноль, но после значительно большего числа повторений деления. Помогите Сереже разобраться с тем, почему это произошло и почему такой результат наблюдался после разного числа повторений деления?

Схема решения. Поскольку при многократном делении Сережа получил ноль, то |a/b|>1. Задача связана с понятием машинного ноля. К пересечению порога машинного ноля приводит многократная операция деления вещественных чисел в силу их представления как (знак, порядок, мантисса). Разница в числе повторений деления связана с тем, что компьютер и калькулятор могут обрабатывать разный диапазон вещественных чисел, у компьютера он несколько больше.

4. На экране монитора нарисована таблица размером на $N \times N$ (на рисунке приведён пример для $N = 5$ и $N = 6$). В некоторых ячейках таблицы записаны целые числа. Найти сумму и максимальное значение элементов, расположенных в закрашенной части таблицы.





Решение. Для обработки закрашенных ячеек будем использовать следующие циклы: номер строки i меняется от 1 до N **div** 2, при этом номер столбца j меняется от 1 до $N - i$. На каждом шаге цикла обрабатываем две ячейки, находящиеся в строках i и $N - i + 1$, т.к. закрашенная часть симметрична. Если количество строк нечётно, то средняя строка не будет обработана, и её надо обработать отдельно, при этом берём i, равное N **div** 2 + 1, j меняется от 1 до $N - i$.

```
алг Таблица
нач
  цел matrix[100, 100], n, i, j, sum, max
  лог first

  ввод n
  если n <= 0 то
    вывод 'Неверное количество строк и столбцов'
  иначе
    для i от 1 до n
    нц
      для i от 1 до n
      нц
        ввод matrix[i, j]
      кц
    кц

    sum = 0
    first = истина
    для i от 1 до n div 2
    нц
      для j от 1 до n – i
      нц
        если matrix[i, j] <> 0 то
          sum = sum + matrix[i, j]
          если first то
            max = matrix[i, j]
            first = ложь
          иначе
            если matrix[i, j] > max то
              max = matrix[i, j]
            всё
          всё
        всё
        если matrix[n – i + 1, j] <> 0 то
          sum = sum + matrix[n – i + 1, j]
          если first то
            max = matrix[n – i + 1, j]
            first = ложь
          иначе
            если matrix[n – i + 1, j] > max то
              max = matrix[n – i + 1, j]
            всё
          всё
        всё
      кц
    кц
    если n mod 2 = 1 то
      i = n div 2 + 1
      для j от 1 до n – i
      нц
        если matrix[i, j] <> 0 то
          sum = sum + matrix[i, j]
          если first то
            max = matrix[i, j]
            first = ложь
          иначе
            если matrix[i, j] > max то
              max = matrix[i, j]
            всё
          всё
        всё
      кц
    всё
    вывод sum, max
  всё
кон
```

5. При шифровании последовательности цифр каждая цифра x заменяется остатком от деления значения многочлена $\mathrm{F}(x) = b \cdot (x^3 + 7x^2 + 3x + a)$ на число 10, где a, b – фиксированные натуральные числа. Разработайте алгоритм определения, при каких значениях a, b указанное преобразование допускает однозначное расшифрование.

Решение. Однозначное расшифрование возможно, если каждой цифре соответствует разный результат. Поскольку нас интересуют остатки от деления, то для a и b достаточно рассмотреть значения от 0 до 9. Для каждой комбинации a и b вычисляем шифр F(x) для всех цифр от 0 до 9 и сравниваем шифры друг с другом. Для упрощения сравнения можно завести массив из 10 элементов с индексами от 0 до 9. В этом массиве индекс будет соответствовать цифре. Далее считаем, сколько раз встречается каждый шифр. Если все шифры разные, то в массиве должны получиться все единицы.

```
алг Расшифрование
нач
  цел a, b, x
  цел s[0..9]
  лог dif

  для a от 0 до 9
  нц
    для b от 0 до 9
    нц
      для x от 0 до 9
      нц
        s[x] = 0
      кц
      для x от 0 до 9
      нц
        f = (b * (x * x * x + 7 * x * x + 3 * x + a)) mod 10
        s[f] = s[f] + 1
      кц
      dif = истина
      x = 0
      пока x < 10 и dif
      нц
        если s[x] <> 1 то
          dif = ложь
        всё
        x = x + 1
      кц
      если dif то
        вывод a, b
      всё
    кц
  кц
кон
```

Если (a_i, b_i) – пара значений, удовлетворяющих условию, то итоговый результат будет иметь вид ($10 \cdot n + a_i$, $10 \cdot n + b_i$), где n – любое натуральное число (в том числе 0).

Если подумать, то можно исключить значение b, равное 0, т.к. при умножении на 0 всегда получается 0. Также можно исключить значение b, равное 5. Если число имеет вид $5 \cdot n$, то остаток от деления на 10 будет равен 0 или 5. Также можно исключить чётные значения b. Если число имеет вид $2 \cdot n$, то остаток от деления на 10 будет чётным. Т.е. в этих случаях не получаются разные шифры для всех цифр.

Если ещё подумать, то можно вычислить, что выражение $x^3 + 7x^2 + 3x$ даёт разные остатки от деления для всех цифр. Следовательно, значение a может быть любым, т.к. оно всегда будет давать одинаковый вклад, и остатки от деления будут оставаться разными. На результат влияет только значение b.

Кроме того, если $x^3 + 7x^2 + 3x + a$ даёт разные остатки от деления, а b равно 1, 3, 7 или 9, то остатки от деления на 10 выражения $b \cdot (x^3 + 7x^2 + 3x + a)$ тоже будут разными.

В итоге получается, что a может быть любым (в том числе больше 9), b имеет вид $10 \cdot n + bb$, где n – любое натуральное число (в том числе 0), а bb есть 1, 3, 7 или 9.