Задания для 10 и 11 классов

1. В теории чисел четвёртая проблема Ландау звучит так: бесконечно ли множество простых чисел вида $n^2 + 1$, где n – натуральное число? Мы не просим Вас подтвердить или опровергнуть гипотезу. Вам предлагается разработать алгоритм для нахождения простых чисел указанного вида.

Решение. Ограничим диапазон поиска некоторым значением k. Построим массив простых чисел в диапазоне от 2 до k, используя решето Эратосфена. Затем для всех n от 1 до $\sqrt{k-1}$ проверим, что число $n^2 + 1$ является простым.

Для использования решета Эратосфена необходимо построить массив чисел в заданном диапазоне от 1 до k. 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 $\sqrt{k}$, 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.

```
алг ПростыеЧислаЛандау()
нач
  цел k, i, j, nums[k]

  ввод k

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

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

  для n от 2 до целая_часть(sqrt(k – 1))
  нц
    если nums[n * n + 1] <> 0 то
      вывод n
    всё
  кц
кон
```

2. 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. Разработайте алгоритм поиска свободных от квадратов чисел в диапазоне от P до Q.

Решение. Можно найти разложение числа N на простые сомножители и проверить сколько раз каждый сомножитель встречается в разложении. Но для этого надо строить массив простых чисел или проверять каждое число на простоту, что достаточно трудоёмко. Проще для каждого числа N в диапазоне от P до Q просмотреть все числа от 2 до $\sqrt{N}$ и проверить, делится ли N на квадрат какого-нибудь из них.

```
алг БесквадратныеЧисла()
нач
  цел p, q, n, k
  лог f

  ввод p, q

  для n от p до q
  нц
    f = истина
    k = 2
    пока k <= целая_часть(sqrt(n)) и f
    нц
      если n mod (k * k) = 0 то
        f = ложь
      всё
      k = k + 1
    кц
    если f то
      вывод n
    всё
  кц
кон
```

3. В теории чисел факториальным простым числом называется простое число, на единицу меньшее или на единицу большее факториала. Вам предлагается разработать алгоритм для нахождения простых чисел указанного вида.

Решение. Попробуем проверить числа на единицу большие или меньшие факториалов всех чисел в диапазоне от 1 до *n*. Значения факториалов расположены достаточно далеко друг от друга и строить массив простых чисел нет смысла, проще проверить на простоту только нужные значения.

```
алг ФакториальныеПростыеЧисла()
нач
  цел n, i, f

  ввод n

  для i от 1 до n
  нц
    f = factorial(i)
    если Простое(f - 1) то
      вывод f - 1
    всё
    если Простое(f + 1) то
      вывод f + 1
    всё
  кц
кон

алг Простое(арг цел N)
нач
  цел i

  если N <= 1 то
    вернуть ложь
  всё
  если N = 2 или  N = 3 или  N = 5 или  N = 7 то
    вернуть истина
  всё
  если N mod 2 = 0 то
    вернуть ложь
  всё
  если N mod 3 = 0 то
    вернуть ложь
  всё

  для i от 5 до целая_часть(sqrt(N)) шаг 6               // Рассматриваем числа, меньшие корня (!) из N
  нц
    если N mod i = 0 то
```

```
      вернуть ложь
    всё
    если N mod (i + 2) = 0 то
      вернуть ложь
    всё
  кц
  вернуть истина
кон

алг factorial(арг цел n)
нач
  цел f

  f = 1
  для i от 1 до n
  нц
    f = f * i
  кц
  вернуть f
кон
```

4. В теории чисел бинарная проблема Гольдбаха звучит так: любое чётное число, начиная с 4, можно представить в виде суммы двух простых чисел. Мы не просим Вас подтвердить или опровергнуть гипотезу. Вам предлагается разработать алгоритм для представления чётных чисел в диапазоне от m до n указанным способом.

Решение. Построим массив простых чисел в диапазоне от 1 до *n*. Далее для каждого числа от *m* до *n* будем перебирать все пары простых чисел из построенного массива, и проверять, не равна ли сумма этой пары текущему числу.

Для использования решета Эратосфена необходимо построить массив чисел в заданном диапазоне от 1 до *k*. 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 $\sqrt{k}$, 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*. Для удобства удалим нулевые элементы массива, чтобы найденные простые числа были расположены последовательно.

```
алг БинарнаяПроблемаГольдбаха()
нач
  цел m, n, k, i, j, p, nums[n]

  ввод m, n

  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 = 0
  для i от 2 до n
```

```
  нц
    если nums[i] <> 0 то
      k = k + 1
      nums[k] = nums[i]
    всё
  кц

  если m < 4 то
    m = 4
  всё
  если m mod 2 = 1 то
    m = m + 1
  всё

  для p от m до n шаг 2
  нц
    для i от 1 до k
    нц
      для j от i до k
      нц
        если nums[i] + nums[j] = p то
          вывод p, " = ", nums[i], " + ", nums[j]
        всё
      кц
    кц
  кц
кон
```

5. В теории чисел число Вудала (Wn) – любое натуральное число вида $n \cdot 2^n - 1$ для некоторого натурального n. Числа Вудала, являющиеся простыми числами, называются простыми числами Вудала. Вам предлагается разработать алгоритм для нахождения простых чисел Вудала в диапазоне от P до Q.

Решение. Для того чтобы уменьшить перебор будем отталкиваться от числа n. Сначала подберём наименьшее значение n такое, чтобы число W_n было больше или равно P. Затем будет увеличивать n на 1, пока число W_n будет меньше или равно Q. Для каждого вычисленного числа W_n будем проверять, является ли оно простым.

```
алг ПростыеЧислаВудала()
нач
  цел p, q, n, w

  ввод p, q

  n = 0
  выполнить
    n = n + 1
    w = n * 2 ^ n – 1
  до w >= p

  пока w <= q
  нц
    если Простое(w) то
      вывод w
    всё
    n = n + 1
    w = n * 2 ^ n – 1
  кц

кон

алг Простое(арг цел N)
нач
  цел i

  если N <= 1 то
    вернуть ложь
  всё
```

```
  если N = 2 или  N = 3 или  N = 5 или  N = 7 то
    вернуть истина
  всё
  если N mod 2 = 0 то
    вернуть ложь
  всё
  если N mod 3 = 0 то
    вернуть ложь
  всё

  для i от 5 до целая_часть(sqrt(N)) шаг 6               // Рассматриваем числа, меньшие корня (!) из N
  нц
    если N mod i = 0 то
      вернуть ложь
    всё
    если N mod (i + 2) = 0 то
      вернуть ложь
    всё
  кц
  вернуть истина
кон
```