

Лучшие задачи отборочного этапа Олимпиады школьников
«Надежда энергетики» по предмету «информатика» в
2018/2019 учебном году.

ЗАДАЧИ ПО ИНФОРМАТИКЕ

1. Разработать алгоритм для вычисления значения выражения

$$R = \sqrt{98 + \sqrt{95 + \sqrt{92 + \dots + \sqrt{5 + \sqrt{2}}}}} . 9 \text{ класс.}$$

Решение. Положим $R = 0$. В цикле для n от 2 до 98 с шагом 3 присваиваем R значение выражения $\sqrt{n + R}$.

алг Корни
нач

цел n
вещ r

$r = 0$

для n от 2 до 98 шаг 3

нц

$r = \text{sqrt}(n + r)$

кц

вывод r

кон

2. Разработать алгоритм для вычисления значения выражения

$$R = \sqrt{98 - \sqrt{95 + \sqrt{92 - \dots - \sqrt{5 + \sqrt{2}}}}} . 10 \text{ класс.}$$

Решение – 1 способ. Положим $R = 0$ и $s = 1$. В цикле для n от 2 до 98 с шагом 3 присваиваем R значение выражения $s \cdot \sqrt{n + R}$, а s умножаем на -1 .

алг Корни
нач

цел n, s
вещ r

$r = 0$

$s = 1$

для n от 2 до 98 шаг 3

нц

$r = s * \text{sqrt}(n + r)$

$s = s * -1$

кц

вывод r

кон

Решение – 2 способ. Положим $R = 0$. В цикле для n от 2 до 92 с шагом 6 присваиваем R значение выражения $-\sqrt{(n+3) + \sqrt{n+R}}$. После цикла присваиваем R значение $\sqrt{98+R}$.

алг Корни
нач

цел n
вещ r

$r = 0$

для n от 2 до 92 шаг 6

нц

$r = -\text{sqrt}((n + 3) + \text{sqrt}(n + r))$

кц

$r = \text{sqrt}(98 + r)$

вывод r

кон

3. На листе бумаги написано Q троек чисел, разделённых пробелом (каждая тройка чисел в новой строке). Рассматривая каждую тройку как коэффициенты уравнений, задающих параболы на плоскости, укажите те параболы, которые имеют одинаковое число пересечений с остальными параболой (не обязательно всеми). 9 класс.

Решение. Пусть a_1, b_1, c_1 – коэффициенты первой параболы, и a_2, b_2, c_2 – коэффициенты второй параболы. Чтобы определить, пересекаются ли параболы, надо выяснить, есть ли значения x , при которых выполняется равенство $a_1 \cdot x^2 + b_1 \cdot x + c_1 = a_2 \cdot x^2 + b_2 \cdot x + c_2$, т.е. $(a_1 - a_2) \cdot x^2 + (b_1 - b_2) \cdot x + (c_1 - c_2) = 0$.

- Если $a_1 - a_2 = 0$ и $b_1 - b_2 = 0$, то параболы не пересекаются (они совпадают при $c_1 - c_2 = 0$ и параллельны при $c_1 - c_2 \neq 0$).
- Если $a_1 - a_2 = 0$, а $b_1 - b_2 \neq 0$, то уравнение $(b_1 - b_2) \cdot x + (c_1 - c_2) = 0$ всегда имеет один корень, т.е. параболы пересекаются.
- Если $a_1 - a_2 \neq 0$, то надо вычислить дискриминант $D = (b_1 - b_2) \cdot (b_1 - b_2) - 4 \cdot (a_1 - a_2) \cdot (c_1 - c_2)$, и если $D \geq 0$, то параболы пересекаются (считаем, что не важно, существует одна точка пересечения или две).

Каждую параболу проверяем на пересечение со всеми остальными параболоми. Сама с собой парабола пересекаться не будет, т.к. в этом случае $a_1 - a_2$ и $b_1 - b_2$ будут равны 0. Для каждой параболы считаем количество её пересечений с другими параболоми (формируем массив), и находим в массиве повторяющиеся значения.

алг Параболы

нач

```
цел q, i, j
вещ a[q], b[q], c[q], d
цел int[q]
```

ввод q

если q <= 0 то

 вывод 'Некорректное значение Q'

иначе

 для i от 1 до q

 нц

 ввод a[i], b[i], c[i]

 кц

 для i от 1 до q

 нц

 int[i] = 0

 для j от 1 до q

 нц

 если a[i] - a[j] = 0 и b[i] - b[j] <> 0 то

 int[i] = int[i] + 1

 иначе

 если a[i] - a[j] <> 0 то

 d = (b[i] - b[j]) * (b[i] - b[j]) - 4 * (a[i] - a[j]) * (c[i] - c[j])

 если d >= 0 то

 int[i] = int[i] + 1

 всё

 всё

 всё

 кц

 кц

 для i от 1 до q - 1

 нц

 для j от i + 1 до q

 нц

 если int[i] = int[j] то

 вывод i, '-ая парабола имеет такое же число пересечений, что и ', j, '-ая парабола'

 всё

 кц

 кц

 всё

кон

4. Арифметический палиндром – положительное число, которое одинаково считывается слева направо и справа налево. Например, 87578 и 123321 это арифметические палиндромы, а 3753 и 81128 нет. Требуется найти количество

способов представить число N как сумму двух арифметических палиндромов. Решения, получаемые перестановкой слагаемых, считать одинаковыми. 9 класс.

Решение. Перебираем числа p от 1 до $N/2$. Если число p является палиндромом, проверяем, является ли палиндромом число $(N - p)$. Если это так, значит мы нашли одно возможное решение.

Для проверки того, что число является палиндромом, надо построить массив цифр числа, деля число на 10 и собирая остатки от деления, и проверить, что получившийся массив является симметричным, т.е. для всех i от 1 до $k/2$ (k – количество цифр в числе) выполняется условие $d_i = d_{k-i+1}$. Можно аналогичным образом сформировать новое число и проверить, равно ли оно исходному.

В принципе, палиндромы можно не искать, а генерировать. Из любого числа из m цифр можно составить два палиндрома, добавив к исходному числу $(m - 1)$ первых цифр в обратном порядке и все m цифр в обратном порядке. Для генерации палиндромов в диапазоне от 10 до L достаточно рассмотреть натуральные числа от 1 до $L/10$. Числа от 1 до 9 считаем палиндромами. Поэтому будем рассматривать числа до максимального значения из 9 и $N/2/10$, и из каждого из них генерировать два палиндрома.

алг СуммаПалиндромов

нач

цел $n, i, p, \text{limit}, \text{count}$

ввод n

если $n \leq 0$ то

 вывод 'Некорректное значение n '

иначе

$\text{count} = 0$

$\text{limit} = 9$

 если $n \text{ div } 20 > \text{limit}$ то

$\text{limit} = n \text{ div } 20$

 всё

 для i от 1 до limit

 нц

$p = \text{Палиндром1}(i)$

 если $p \leq n \text{ div } 2$ то

 если $(n - p) = \text{ОбратноеЧисло}(n - p)$ то

$\text{count} = \text{count} + 1$

 всё

 всё

$p = \text{Палиндром2}(i)$

 если $p \leq n \text{ div } 2$ то

 если $(n - p) = \text{ОбратноеЧисло}(n - p)$ то

$\text{count} = \text{count} + 1$

 всё

 всё

 кц

всё

кон

алг Палиндром1(арг цел i)

нач

цел res

$\text{res} = i$

$i = i \text{ div } 10$

пока $i <> 0$

нц

$\text{res} = \text{res} * 10 + i \text{ mod } 10$

$i = i \text{ div } 10$

кц

вернуть res

кон

алг Палиндром2(арг цел i)

нач

цел res

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

```
res = i
пока i <> 0
нц
  res = res * 10 + i mod 10
  i = i div 10
кц
вернуть res
кон
```

```
алг ОбратноеЧисло(арг цел i)
нач
  цел res

  res = 0
  пока i <> 0
  нц
    res = res * 10 + i mod 10
    i = i div 10
  кц
  вернуть res
кон
```

5. На бумаге нарисован ряд из M клеток. В клетке либо пусто, либо записано целое число. Найти минимальное число, встречающееся ровно 4 раза. *9 класс.*

Решение. Для каждого числа надо определить количество его вхождений. Можно пойти двумя путями.

1. Сформировать массив, в котором индексами служат числа исходного массива, а значениями – количества вхождений каждого числа. В этом случае надо как-то ограничить числа, входящие в исходный массив.
2. Сформировать два новых массива: в одном будут числа исходного массива, во втором – количества их вхождений. Для каждого числа надо будет проверять, встречается ли оно уже в первом массиве, если так – увеличиваем количество вхождений, иначе добавляем в первый массив это число, а во второй – единицу.

В любом случае далее ищем числа, которые встречаются 4 раза, переносим их ещё в один массив и находим в нём минимум.

```
алг Минимум4
нач
  цел m, arr[m], arr4[m]
  цел entries[100] // Если исходные числа лежат в диапазоне от 1 до 100
  цел min, n, i

  ввод m
  если m < 0 то
    вывод 'Некорректное значение m'
  иначе
    для i от 1 до m
    нц
      ввод arr[i]
    кц

    для i от 1 до 100
    нц
      entries[i] = 0
    кц

    для i от 1 до m
    нц
      если arr[i] <> 0 то
        entries[arr[i]] = entries[arr[i]] + 1
      всё
    кц

  n = 0
  для i от 1 до 100
  нц
    если entries[i] = 4 то
      n = n + 1
```

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

```
arr4[n] = arr[i]
всё
кц

если n = 0 то
вывод 'Нет элементов, входящих 4 раза'
иначе
min = arr4[1]
для i от 2 до n
нц
если arr4[i] < min то
min = arr4[i]
всё
кц
вывод 'Минимальное число, входящее 4 раза, равно ', min
всё
кц
кон
```

алг Минимум4

нач

цел m, arr[m], arr4[m]

цел entries[2][m]

цел min, n, k, i, j, p

ввод m

если m < 0 то

вывод 'Некорректное значение'

иначе

для i от 1 до m

нц

ввод arr[i]

кц

k = 0

для i от 1 до m

нц

если arr[i] <> 0 то

p = 0

j = 1

пока j <= k и p = 0

нц

если arr[i] = entries[1][j] то

p = j

всё

j = j + 1

кц

если p <> 0 то

entries[2][p] = entries[2][p] + 1

иначе

k = k + 1

entries[1][k] = arr[i]

entries[2][k] = 1

всё

кц

n = 0

для i от 1 до k

нц

если entries[2][i] = 4 то

n = n + 1

arr4[n] = entries[1][i]

всё

кц

если n = 0 то

вывод 'Нет элементов, входящих 4 раза'

иначе

min = arr4[1]

для i от 2 до n

нц

если arr4[i] < min то

min = arr4[i]

всё

кц

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

```
    вывод 'Минимальное число, входящее 4 раза, равно ', min
  всё
всё
кон
```

6. На бумаге нарисована таблица размером $M \times N$ клеток. В клетке либо пусто, либо записано целое число. Справа от таблицы есть пустая таблица размером $M \times N$ клеток. В правую таблицу надо записать все числа из исходной таблицы, расположив их в порядке убывания. Начинать с угла $(1, 1)$, двигаясь вертикально. Пустые клетки расположить в начале таблицы по строкам. *10 класс.*

Решение. Перенесём числа из таблицы в одномерный массив, начиная с элемента $(1, 1)$ и просматривая таблицу по столбцам. Также подсчитаем количество пустых клеток K . Упорядочим массив каким-либо известным способом. При переносе значений из массива в новую таблицу надо оставить пустыми первые $K \mathbf{div} N$ строк, а также $K \mathbf{mod} N$ первых элементов строки с номером $K \mathbf{div} N + 1$. Далее упорядоченные числа из массива надо записать в оставшиеся клетки по столбцам. В столбцах от 1 до $K \mathbf{mod} N$ заполняем строки от $K \mathbf{div} N + 2$ до M , а в столбцах от $K \mathbf{mod} N + 1$ до N заполняем строки от $K \mathbf{div} N + 1$ до M . Если $K \mathbf{mod} N = 0$, то первый цикл для столбцов от 1 до $K \mathbf{mod} N$ получается вырожденным, а второй цикл для столбцов от $K \mathbf{mod} N + 1$ до N заполняет нужное количество полных строк.

```
алг Таблица
нач
  цел m, n, k, p, i, j
  цел table[m, n], newtable[m, n]

  ввод m, n
  если m <= 0 или n <= 0 то
    вывод 'Неверные размеры таблицы'
  иначе
    для i от 1 до m
      нц
        для j от 1 до n
          нц
            ввод table[i, j]
          кц
        кц

    p = 0
    для j от 1 до n
      нц
        для i от 1 до m
          нц
            если table[i, j] <> 0 то
              p = p + 1
              arr[p] = table[i, j]
            всё
          кц
        кц
    k = m * n - p

    QuickSort(arr, 1, p)

    для i от 1 до k div n
      нц
        для j от 1 до n
          нц
            newtable[i, j] = 0
          кц
        кц
    для j от 1 до k mod n
      нц
        newtable[k div n + 1, j] = 0
      кц

    p = 0
    для j от 1 до k mod n
      нц
        для i от k div n + 2 до m
          нц
```

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

```

    p = p + 1
    newtable[i, j] = arr[p]
кц
кц
для j от k mod n + 1 до n
нц
    для i от k div n + 1 до m
    нц
        p = p + 1
        newtable[i, j] = arr[p]
    кц
кц

для i от 1 до m
нц
    для j от 1 до n
    нц
        вывод newtable[i, j]
    кц
кц
всё
кон

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

    если n2 - n1 = 1 то
        если 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 x[i] < k
                нц
                    i = i + 1
                кц
                пока 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)
            QuickSort(x, i, n2)
        всё
    всё
кон

```

7. Дана арифметическая прогрессия, члены которой выписаны в одну строку: 11213141... Вывести k -ю цифру в получившейся строке. 10 класс.

Решение. Положим первое число равным 11. Положим длину числа ln равной 2 и длину последовательности ls тоже равной 2. В цикле увеличиваем число на 10, определяем его длину и прибавляем эту длину к общей длине последовательности ls . Для определения длины числа можно подсчитывать количество его цифр, но это потребует нескольких операций деления, что не практично. Поэтому будем сравнивать каждое число с пороговым значением h , изначально равным 100. Если число меньше порогового значения, значит, его длина равна текущему значению ln . Если же число

становится больше или равным значению h , то значение ln надо увеличить на 1, а h умножить на 10.

Продолжаем цикл, пока длина последовательности ls меньше k . Когда ls станет равной или больше k , значит, нужная нам цифра есть $(ls - k + 1)$ цифра с конца текущего числа. Для вычисления этой цифры надо поделить текущее число на 10^{ls-k} и после этого взять остаток от деления на 10.

```

алг Последовательность
нач
    цел k, num, ln, ls, h

    ввод k
    если k <= 0 то
        вывод 'Некорректное значение'
    иначе
        num = 11
        ln = 2
        ls = 2
        h = 100
        пока ls < k
            нц
                num = num + 10
                если num >= h то
                    ln = ln + 1
                    h = h * 10
                всё
                ls = ls + ln
            кц
        вывод (num div (10 ^ (ls - k))) mod 10
    всё
кон
    
```

8. В квадратной таблице размера $n \times n$ записаны целые числа. В таблице проведена диагональ из угла $(n, 1)$ в угол $(1, n)$. Найти минимум среди сумм модулей элементов диагоналей, параллельных проведённой диагонали таблицы. *10 класс.*

Решение. Если i – номер строки, то номер столбца для элемента побочной диагонали вычисляется как $n - i + 1$. Найдём сумму модулей элементов побочной диагонали и используем её как начальное значение минимума. Далее мы имеем по $n - 1$ диагоналей выше и ниже побочной, причём, они симметричны относительно побочной диагонали. Напишем цикл для k от 2 до n для перебора диагоналей, для k -ой диагонали ниже побочной номер строки i меняется от k до n , а номер столбца j вычисляется как $n + k - i$, для k -ой диагонали выше побочной номер строки есть $n + 1 - j$, а номер столбца есть $n + 1 - i$. Находим суммы модулей элементов диагоналей, сравниваем их с текущим минимумом, при необходимости меняем минимум.

```

алг Диагонали
нач
    цел n, x[n, n], min, s, s1, s2, k, i, j

    ввод n
    если n <= 0 то
        вывод 'Некорректный размер таблицы'
    иначе
        для i от 1 до n
            нц
                для j от 1 до n
                    нц
                        ввод x[i, j]
                    кц
                кц
            кц

            s = 0
            для i от 1 до n
                s = s + abs(x[i, n + 1 - i])
            min = s
            для k от 2 до n
                нц
                    s1 = 0
                    
```

```
s2 = 0
для i от k до n
нц
    j = n + k - i
    s1 = s1 + abs(x[i, j])
    s2 = s2 + abs(x[n + 1 - j, n + 1 - i])
кц
если s1 < min то
    min = s1
если s2 < min то
    min = s2
кц

вывод min
всё
кон
```

9. Для заданного натурального числа N в арифметическом выражении

$$(\dots (((1 ? 2) ? 3) ? 4) \dots) ? N,$$

вместо каждого знака $?$ вставить знак одной из четырёх арифметических операций $+$, $-$, $*$, $/$ так, чтобы результат вычислений равнялся заданному числу Y . При делении дробная часть в частном отбрасывается. Необходимо учитывать приоритет операций. Достаточно найти одно решение. Вывести последовательность арифметических операций. Если решение найти невозможно, вывести фразу «НЕТ РЕШЕНИЯ». *11 класс.*

Решение. Для решения этой задачи необходимо найти все перестановки знаков арифметических операций с повторениями. Создадим массив из $(N - 1)$ элементов. Каждый элемент этого массива будет хранить знак операции в виде одного из чисел 0, 1, 2 или 3. Нумерация знаков начинается с 1. Изначально значение всех элементов массива равно 0, что соответствует всем знакам «плюс». Для решения задачи потребуется три вспомогательных алгоритма: вычисления выражения, вывода выражения и перехода к следующей комбинации знаков.

В цикле выполняем следующие действия:

- вычисляем значение выражения;
- если оно равно искомому, выводим выражение и его значение, прерываем цикл;
- переходим к следующей комбинации знаков.

Цикл прекращается, когда алгоритм перехода к следующей комбинации знаков возвращает 1, что соответствует переносу за границу числа при прибавлении 1 к максимальному возможному значению. Это будет означать, что не найдено ни одного решения.

Для вычисления выражения положим значения результата r равным 1 (значение первого аргумента выражения), далее в цикле для i от 1 до $N - 1$ вычисляем $r = r s_i (i + 1)$, где s_i – i -ый знак операции в выражении.

Для вывода выражения напишем следующую рекурсивную функцию:

- если N равно 2, то выводим $1 s_1 2$, где s_1 – первый знак операции в выражении;
- иначе:
 - выводим открывающую скобку;
 - с помощью рекурсивного вызова выводим комбинацию, которая содержит первые $(N - 1)$ числа и знаки операций между ними;
 - выводим закрывающую скобку, последний знак операции и число N .

Поскольку знаки операций кодируются числами от 0 до 3, переход к следующей комбинации эквивалентен прибавлению 1 к числу, представленному в системе счисления с основанием 4. Для перехода к следующей комбинации положим переменную $carry$ равной 1 и в цикле для i от $N - 1$ до 1 прибавляем переменную $carry$ к

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

i -ому элементу массива, при этом, если i -ый элемент массива становится равным 4, то ему присваивается 0, а значение переменной *carry* сохраняется, в противном случае переменной *carry* присваивается 0 и цикл прекращается. Когда массив представляет максимальное возможное значение, прибавление 1 переведёт все элементы массива в 0, при этом переменная *carry* останется равной 1, что соответствует переносу за границу числа. Это будет означать, что были перебраны все возможные комбинации.

алг Выражение

нач

```
цел константа plus = 0, minus = 1, multiplication = 2, division = 3
цел константа max = 4
симв signs[] = {'+', '-', '*', '/'}
цел expr[1000]
цел n, y, i
лог found
```

ввод n, y

если n <= 0 то

 вывод 'Некорректное значение n'

иначе

 для i от 1 до n - 1

 нц

 expr[i] = 0

 кц

found = ложь

повторять

 если ExpressionValue(n) = y то

 вывод y, ' = '

 PrintCombination(n)

 found = истина

 всё

до NextCombination(n) = 1 или found

если не found то

 вывод 'Нет решения'

всё

конец

алг Eval(арг цел x, sign, y)

нач

 если sign = plus то

 вернуть x + y

 если sign = minus то

 вернуть x - y

 если sign = multiplication то

 вернуть x * y

 если sign = division то

 вернуть x / y

конец

алг ExpressionValue(арг цел n)

нач

 цел r, i

 r = 1

 для i от 1 до n - 1

 нц

 r = Eval(r, expr[i], i + 1)

 кц

 вернуть r

конец

алг PrintCombination(арг цел n)

нач

 если n = 2 то

 вывод '1 ', signs[expr[1]], ' 2'

 иначе

 вывод '('

 PrintCombination(n - 1)

 вывод ')', signs[expr[n - 1]], n

 всё

конец

алг NextCombination(арг цел n)

```

нач
цел carry, i

carry = 1
i = n - 1
пока i >= 1 и carry = 1
нц
  expr[i] = expr[i] + carry
  если expr[i] = max то
    expr[i] = 0
  иначе
    carry = 0
  всё
  i = i - 1
кц
вернуть carry
кон
    
```

10. Для заданного натурального числа N в арифметическом выражении

$$1 ? 2 ? 3 ? \dots ? N$$

вместо каждого знака $?$ вставить знак одной из четырёх арифметических операций $+$, $-$, $*$, $/$ так, чтобы результат вычислений равнялся заданному числу Y . При делении дробная часть в частном отбрасывается. Необходимо учитывать приоритет операций. Достаточно найти одно решение. Вывести последовательность арифметических операций. Если решение найти невозможно, вывести фразу «НЕТ РЕШЕНИЯ». *11 класс.*

Решение. Для решения этой задачи необходимо найти все перестановки знаков арифметических операций с повторениями. Создадим массив из $(N - 1)$ элементов. Каждый элемент этого массива будет хранить знак операции в виде одного из чисел 0, 1, 2 или 3. Нумерация знаков начинается с 1. Изначально значение всех элементов массива равно 0, что соответствует всем знакам «плюс». Для решения задачи потребуется три вспомогательных алгоритма: вычисления выражения, вывода выражения и перехода к следующей комбинации знаков.

В цикле выполняем следующие действия:

- вычисляем значение выражения;
- если оно равно искомому, выводим выражение и его значение, прерываем цикл;
- переходим к следующей комбинации знаков.

Цикл прекращается, когда алгоритм перехода к следующей комбинации знаков возвращает 1, что соответствует переносу за границу числа при прибавлении 1 к максимальному возможному значению. Это будет означать, что не найдено ни одного решения.

Для вычисления выражения положим значения результата r равным 1 (значение первого аргумента выражения), далее в цикле для i от 1 до $N - 2$ делаем следующее:

- если i -ый знак операции есть «плюс» или «минус», а $(i + 1)$ -ый знак операции есть «умножение» или «деление», то:
 - положим переменную sr равной $i + 1$;
 - в цикле для j от $i + 1$ до $N - 1$ пока j -ый знак операции есть «умножение» или «деление» вычисляем $sr = sr s_j (j + 1)$, где s_j – j -ый знак операции в выражении;
 - затем вычислим $r = r s_i sr$ и положим i равным $j - 1$;
- иначе вычисляем $r = r s_i (i + 1)$.

После цикла проверяем, что $i = N - 1$ и, если это так, ещё раз вычисляем $r = r s_i (i + 1)$.

Для вывода выражения в цикле для i от 1 до $N - 1$ выводим число i и i -ый знак операции в выражении, затем выводим число N .

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

Поскольку знаки операций кодируются числами от 0 до 3, переход к следующей комбинации эквивалентен прибавлению 1 к числу, представленному в системе счисления с основанием 4. Для перехода к следующей комбинации положим переменную *carry* равной 1 и в цикле для *i* от $N - 1$ до 1 прибавляем переменную *carry* к *i*-ому элементу массива, при этом, если *i*-ый элемент массива становится равным 4, то ему присваивается 0, а значение переменной *carry* сохраняется, в противном случае переменной *carry* присваивается 0 и цикл прекращается. Когда массив представляет максимальное возможное значение, прибавление 1 переведёт все элементы массива в 0, при этом переменная *carry* останется равной 1, что соответствует переносу за границу числа. Это будет означать, что были перебраны все возможные комбинации.

алг Выражение

нач

цел константа plus = 0, minus = 1, multiplication = 2, division = 3

цел константа max = 4

симв signs[] = {'+', '-', '*', '/'}

цел expr[1000]

цел n, y, i

лог found

ввод n, y

если $n \leq 0$ то

 вывод 'Некорректное значение'

иначе

 для *i* от 1 до $n - 1$

 нц

 expr[*i*] = 0

 кц

 found = ложь

 повторять

 если ExpressionValue(*n*) = *y* то

 вывод *y*, '=',

 PrintCombination(*n*)

 found = истина

 всё

 до NextCombination(*n*) = 1 или found

 если не found то

 вывод 'Нет решения'

 всё

всё

кон

алг Eval(арг цел *x*, sign, *y*)

нач

 если sign = plus то

 вернуть $x + y$

 если sign = minus то

 вернуть $x - y$

 если sign = multiplication то

 вернуть $x * y$

 если sign = division то

 вернуть x / y

кон

алг ExpressionValue(арг цел *n*)

нач

 цел *r*, *i*, *sr*, *j*

r = 1

 для *i* от 1 до $n - 2$

 нц

 если (expr[*i*] = plus или expr[*i*] = minus) и (expr[*i* + 1] = multiplication или expr[*i* + 1] = division) то

sr = *i* + 1

j = *i* + 1

 пока $j \leq n - 1$ и (expr[*j*] = multiplication или expr[*j*] = division)

 нц

sr = Eval(*sr*, expr[*j*], *j* + 1)

j = *j* + 1

 кц

r = Eval(*r*, expr[*i*], *sr*)

i = *j* - 1

 иначе

Олимпиада школьников «Надежда энергетики». Отборочный этап. Решения.

```
    r = Eval(r, expr[i], i + 1)
  всё
кц
если i = n - 1 то
  r = Eval(r, expr[i], i + 1)
всё
вернуть r
кон
```

```
алг PrintCombination(арг цел n)
нач
  цел i

  для i от 1 до n - 1
  нц
    вывод i, signs[expr[i]]
  кц
вывод n
кон
```

```
алг NextCombination(арг цел n)
нач
  цел carry, i

  carry = 1
  i = n - 1
  пока i >= 1 и carry = 1
  нц
    expr[i] = expr[i] + carry
    если expr[i] = max то
      expr[i] = 0
    иначе
      carry = 0
    всё
    i = i - 1
  кц
  вернуть carry
кон
```