**Готуємось до олімпіади з інформатики 2017-1**

*(класичні задачі початкового рівня)*

**Задача "Палиндром":** Определить, палиндром ли *слово*, введенное с клавиатуры (палиндром читается одинаково слева направо и справа налево).

**Идея решения:** Во введенной строке необходимо проверить - равны ли первый и последний символы, второй и предпоследний и т.д. (используя зависимость уменьшающейся переменной X в теле *цикла* от увеличивающегося значения счетчика *цикла* i).

Обратите внимание, что *тело цикла* выполняется n/2 раз (за один проход сравниваются 2 символа).

**Задача:** Найти делители введенного с клавиатуры числа N.

**Идея решения:** Во всех приведенных ниже задачах (набора для практики) необходимо воспользоваться формулой n mod i=0 (проверить - равен ли *остаток* от деления нулю). Cчетчик *цикла* "перебирает" возможные делители числа n.

* **Простое число** - число, большее чем 1 и делящееся только лишь на 1 и на само себя.
* **Совершенное число** - число, равное сумме своих делителей (не считать делителем числа само число).
* **Дружественные числа** - пара чисел, каждое из которых равно сумме правильных делителей другого.
* **Период строки** - повторяющаяся часть строки

Ниже приведены задачи из раздела "Теория чисел" с разбором решений.

**Задача:** Проверить - простое ли число, введенное с клавиатуры.

**Задача:** В диапазоне от А до В вывести все простые числа.

**Задача:** Проверить - совершенное ли число, введенное с клавиатуры.

**Задача:** На диапазоне от А до В вывести все совершенные числа.

**Задача:** Проверить, являются ли два числа, введенных с клавиатуры дружественными.

**Задача:** Ввести строку. Найти ее период.

**Идея решения:** Необходимо найти К (количество символов в строке), определить делители К и сравнить вырезки из строки по R символов (R-делитель К)

**Типовые алгоритмы обработки одномерных массивов**

В данной теме будут рассматриваться такие типовые алгоритмы обработки одномерных массивов:

* Заполнение, вывод элементов массива
* Сумма, произведение элементов
* Выбор по условию
* Максимальный (минимальный) элемент
* Вставка, удаление элементов
* Инвертирование (изменения порядка следования элементов заданного массива на обратный)

**Задача:** На плоскости изображено N прямоугольников ([рис. 2.1](http://www.intuit.ru/studies/courses/2293/593/lecture/12749?page=1#image.2.1)). Каждый прямоугольник задан координатами левой нижней и правой верхней вершин. Определить, имеют ли прямоугольники общую площадь .



**Рис. 2.1.**

**Идея решения:**

Если:

* максимальная координата по оси Х левых нижних вершин прямоугольников будет меньше минимальной координаты правых верхних вершин и …
* …максимальная координата по оси У левых нижних вершин прямоугольников будет меньше минимальной координаты правых верхних вершин, то …
* …общая площадь есть.

В задаче необходимо использовать типовой алгоритм нахождения МАКСИМАЛЬНОГО (МИНИМАЛЬНОГО) ЭЛЕМЕНТА МАССИВА.

Для вычисления общей площади необходимо найти произведение разности:

* максимальной координаты по оси Х левых нижних вершин прямоугольников и минимальной координаты правых верхних вершин и …

…максимальной координаты по оси У левых нижних вершин прямоугольников и минимальной координаты правых верхних вершин.

**Задача: Латинским квадратом** называется массив, в строках и столбцах которого нет одинаковых элементов. Вывести на экран латинский квадрат размером NxN.

**Идея решения:** Заполнить 1 строку квадратного массива (NxN) числами от 1 до N. Вторая строка массива получается путем циклического сдвига элементов первой строки и т. д. ([табл. 2.2](http://www.intuit.ru/studies/courses/2293/593/lecture/12749?page=1#table.2.2)). Циклический сдвиг можно реализовать, используя типовой алгоритм ВСТАВКИ-УДАЛЕНИЯ (в зависимости от направления циклического сдвига).

|  |
| --- |
| Таблица 2.2. Заполнение латинского квадрата путем циклического сдвига элементов |
| 1 | 2 | 3 | 4 | 5 |
| 5 | 1 | 2 | 3 | 4 |
| 4 | 5 | 1 | 2 | 3 |
| 3 | 4 | 5 | 1 | 2 |
| 2 | 3 | 4 | 5 | 1 |

**Задачи "Разбор числа на цифры"**

**Идея решения:** Во всех задачах данной группы число необходимо ввести в строковую ячейку. Затем, применив типовой алгоритм "РАЗБОРА" ЧИСЛА НА ЦИФРЫ, произвести необходимые действия. Ячейка, в которой накапливается произведение в цикле, должна быть предварительно заполнена единицей.

Ввести число. Найти сумму, произведение цифр числа.

Ввести число. Найти количество нулевых цифр в числе

Определить, является ли число, введенное с клавиатуры числом Армстронга. *Число Армстронга:* сумма цифр этого числа, возведенных в n-ую степень (n - количество цифр числа) равна самому числу.

**Идея решения:** Число необходимо ввести в строковую ячейку. Аналогично предыдущей задаче необходимо применить типовой алгоритм РАЗБОРА ЧИСЛА НА ЦИФРЫ. В программе на Паскале возведение числа в n-ую степень реализовано в цикле.

Найти цифровой корень числа. **Цифровой корень** числа получается при сложении цифр числа, затем при сложении цифр вновь полученного числа и так до тех пор, пока в сумме не будет получена одна цифра.

Перевести число из одной системы счисления в другую.

**Идея решения:** перевод числа в десятичную систему счисления базируется на типовом алгоритме "РАЗБОРА" ЧИСЛА НА ЦИФРЫ. Перевод из десятичной системы счисления не опирается на рассмотренные выше типовые алгоритмы, но мы приведем и его.

Вывести на экран самое длинное и самое короткое слово в предложении.

**Идея решения:** Введенное с клавиатуры предложение, воспользовавшись типовым алгоритмом "РАЗБОРА" ПРЕДЛОЖЕНИЯ НА СЛОВА помещаем в массив. Затем, применив типовой алгоритм ПОИСКА МАКСИМАЛЬНОГО ЭЛЕМЕНТА найдем самое длинное и самое короткое слово в предложении.

Ввести предложение. Выдать его на экран, поменяв порядок следования слов в предложении.

**Идея решения:** Введенное с клавиатуры предложение, воспользовавшись типовым алгоритмом "РАЗБОРА" ПРЕДЛОЖЕНИЯ НА СЛОВА помещаем в массив. Выводим массив с конца. Для организации хранения в массиве слов в обратном порядке необходимо воспользоваться зависимостью УМЕНЬШАЮЩЕЙСЯ ПЕРЕМЕННОЙ ОТ УВЕЛИЧИВАЮЩЕГОСЯ СЧЕТЧИКА ЦИКЛА и поменять порядок следования слов в массиве (проделайте это самостоятельно).

Ввести предложение. Выдать его на экран, изменив порядок следования букв в каждом слове, оставив порядок следования слов в предложении прежним.

**Идея решения:** Введенное с клавиатуры предложение, воспользовавшись типовым алгоритмом "РАЗБОРА" ПРЕДЛОЖЕНИЯ НА СЛОВА помещаем в массив. Инвертируем содержимое каждого элемента массива. В приведенной ниже программе на Паскале в массиве хранится предложение с обратным порядком слов. Изменить порядок слов можно, воспользовавшись зависимостью УМЕНЬШАЮЩЕЙСЯ ПЕРЕМЕННОЙ ОТ УВЕЛИЧИВАЮЩЕГОСЯ СЧЕТЧИКА ЦИКЛА.

Упражнения

* Найти произведение четных, сумму отрицательных, количество нулевых элементов одномерного массива размерностью 10, заполненного с клавиатуры.
* Удалить максимальный элемент из одномерного массива размерностью 10, заполненного с клавиатуры. Вставить после минимального элемента ноль.
* Ввести число. Найти произведение четных цифр в нем.
* Ввести число в десятичной системе счисления. Определить, чего больше - нулей или единиц в его двоичном представлении?
* Ввести предложение. Найти, сколько слов нем начинается и заканчивается одной и той же буквой.

**Задачи "Подпоследовательности"**

1. В последовательности чисел выделить **все подпоследовательности подряд идущих чисел.**

**Идея решения**: Ряд чисел СОРТИРУЕМ. Перебираем элементы массива, сравнивая два соседних элемента. Если разница между ними не равна единице, то начинаем печать новой подпоследовательности.

1. В последовательности чисел найти и вывести **подпоследовательность подряд идущих чисел наибольшей длины.**

**Идея решения**: Ряд чисел СОРТИРУЕМ. Перебираем элементы массива, сравнивая два соседних элемента. Если разница между ними равна единице, значит очередное число входит в текущую подпоследовательность. В ячейке К накапливаем ее длину. Применив типовой алгоритм поиска МАКСИМАЛЬНОГО ЭЛЕМЕНТА находим подпоследовательность наибольшей длины.

**Задача:** Ввести предложение. Поменять порядок слов в нем, расположив их **в порядке увеличения букв** в словах.

**Идея решения:** Применив типовой алгоритм "РАЗБОРА" ПРЕДЛОЖЕНИЯ НА СЛОВА, помещаем каждое слово в элемент массива. Далее СОРТИРУЕМ массив слов, сравнивая количество букв в словах.

Обработку двумерного массива можно условно разделить на три группы:

* Обработка всего массива;
* Обработка отдельно по строкам и столбцам;
* Обработка относительно диагоналей.

**Задача 1**: Написать программу для поиска всех простых чисел (до числа N).

**Дополнительные сведения:** Рассмотрим *алгоритм* поиска простых чисел на диапазоне от 1 до N методом "**Решето Эратосфена**". Проиллюстрируем этот метод:

* *1 шаг:* берем "2", отмечаем серым цветом. Вычеркиваем все числа, кратные двум.
* *2 шаг:* берем следующее за "2" невычеркнутое число ("3"), отмечаем серым цветом. Вычеркиваем все числа, кратные трем и т.д.
* В "решете" остались числа, отмеченные серым цветом. Это простые числа.



**Рис. 6.1.**

**Идея решения:**

* Заполняем одномерный массив А подряд идущими числами до N (например: N=22) включительно (таблица ниже):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |

* Заполняем элементы массива Flag флажками-единицами, если соответствующие элементы массива А кратны "2" ([рис. 6.2](http://www.intuit.ru/studies/courses/2293/593/lecture/12756?page=1#image.6.2)):



**Рис. 6.2.**

* Затем заполняем элементы массива Flag флажками-единицами, если соответствующие элементы массива А кратны "3" ([рис. 6.3](http://www.intuit.ru/studies/courses/2293/593/lecture/12756?page=1#image.6.3)):



**Рис. 6.3.**

* и т.д.
* В "решете" (массиве А) остались числа, отмеченные серым цветом ([рис. 6.4](http://www.intuit.ru/studies/courses/2293/593/lecture/12756?page=1#image.6.4)):



**Рис. 6.4.**

Для сокращения количества шагов достаточно перебирать элементы массива А до половины (т.к. максимальный делитель n - это n/2).

**Задача 2:** В одномерном массиве, заполненном целыми числами подсчитать число различных элементов.

**Идея решения:** сравнивая очередной элемент массива А с остальными элементами, заполняем единицей соответствующий повторяющемуся элементу элемент массива Flag. Количество нулевых элементов в массиве Flag и будет равно количеству различных элементов массива А.

**Задача 3:** Вывести элемент, встречающийся в одномерном массиве чаще других.

**Идея решения:** При сравнении элементов массива А находим повторяющиеся элементы. *Массив* Flag заполняем количеством повторений элемента массива А. Затем, применив типовой *алгоритм* ПОИСКА МАКСИМАЛЬНОГО ЭЛЕМЕТА МАССИВА находим позицию элемента массива А, встречающегося чаще других.

**Задача 1:** В строке, содержащей арифметическое *выражение* проверить, правильно ли расставлены скобки. В случае лишних скобок - не считать неправильным расстановку скобок (например: ((а+в)) -скобки расставлены верно).

**Идея решения:**

1. Допустим, дано арифметическое выражение: (а + в ((4а - 8) в + 3) - 15в)
2. Резервируем два массива: А$ и Flag. "Разбираем" строку, в которой хранится арифметическое выражение (применив типовой алгоритм "РАЗБОРА" СТРОКИ НА СИМВОЛЫ), помещая каждый символ в элемент массива А$.

Массив Flag заполняем флажками:

* + "1" - если элемент массива А$ - открывающаяся скобка;
	+ "-1" - если элемент массива А$ - закрывающаяся скобка ([рис. 7.1](http://www.intuit.ru/studies/courses/2293/593/lecture/12758?page=1#image.7.1)).



**Рис. 7.1.**

1. Суммируем элементы массива Flag. Если в процессе перебора элементов сумма станет отрицательной, значит скобки расставлены неверно. Итоговая сумма должна быть равна 0.

**Задача 2:** В картинной галерее работают сторожа. Для каждого сторожа известно время прихода на работу и время ухода. Определить, всегда ли галерея охраняется.

**Идея решения:**Пример ([табл. 7.1](http://www.intuit.ru/studies/courses/2293/593/lecture/12758?page=1#table.7.1)):

|  |
| --- |
| Таблица 7.1. |
|  | **Время прихода** | **Время ухода** |
| 1 сторож | 8.00 | 12.00 |
| 2 сторож | 11.00 | 16.00 |
| 3 сторож | 15.00 | 19.30 |
| 4 сторож | 20.00 | 23.50 |

1. Заполняем массивы:

А: временем прихода и ухода сторожей;

Flag:

* + "1" - если соответствующий элемент массива А - время прихода сторожа;
	+ "-1" - если элемент массива А - время ухода сторожа;



**Рис. 7.2.**

1. СОРТИРУЕМ массив А, одновременно переставляя элементы массива flag:



**Рис. 7.3.**

1. Суммируем элементы второго массива. Если текущая сумма обнулилась (но конец массива не достигнут), то, значит, галерея осталась без охраны (в 19.30 сторож ушел, а смена еще не пришла).



**Рис. 7.4.**

**Задача 3:** N отрезков на координатной *прямой* заданы координатами своих концов. Определить количество **связных областей**.

**Идея решения:**

1. Заполняем массивы:

А: координатами начала и конца отрезков;

Flag:

* + "1" - если соответствующий элемент массива А - координата начала отрезка;
	+ "-1" - если элемент массива А - координата конца отрезка;
1. СОРТИРУЕМ массив А, одновременно переставляя элементы массива Flag.
2. Суммируем элементы массива Flag. Если текущая сумма обнулилась (но конец массива не достигнут), то, значит, получена одна связная область. Количество обнулений и будет количеством связных областей.

**Задача 1: "Арифметический квадрат".**

Сколько путей ведет из клетки (1, 1) в клетку (N, M), с условием, что двигаться можно только вниз или вправо?

**Идея решения:**

1. Из клетки (1,1) в любую клетку первой строки и любую клетку первого столбца ведет только один путь (по первой строке - вправо и по первому столбцу - вниз). Заполняем единицами (обозначающими единственный путь) строку и столбец ([рис. 8.1](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=1#image.8.1)).



**Рис. 8.1.**

1. В клетку (2,2) из клетки (1,1) ведут 2 пути ([рис.8.2](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=1#image.8.2)):



**Рис. 8.2.**

1. В клетку (2,3) ведут 3 пути ([рис.8.3](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=1#image.8.3)):



**Рис. 8.3.**

1. и т.д.
2. Количество путей, ведущих в клетку из клетки (1,1) зависит от значений верхнего и левого элементов массива ([рис.8.4](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=1#image.8.4)):

**Итого:** в клетку (4, 4) ведет 20 путей.

**Задача 2**: Выведите на экран "**Треугольник Паскаля**" ([рис.8.4](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=1#image.8.4)):



**Рис. 8.4.**

**Идея решения:**

*Треугольник Паскаля* расположен выше и на побочной диагонали Арифметического квадрата:



**Задача 3:** Раскрыть скобки в выражении: (А+В)n (n ввести с клавиатуры).

**Дополнительные сведения:**

"*Бином Ньютона*":



и т. д.

**Идея решения:**

* Коэффициенты при слагаемых находятся НА ОСНОВАНИИ *ТРЕУГОЛЬНИКА ПАСКАЛЯ* (на побочной диагонали Арифметического квадрата, размерностью (n+1)x(n+1)).
* Степень А уменьшается, степень В увеличивается от 0 до n.

**Задача 4:** Найти кратчайший *путь* в двумерном массиве из клетки 1,1 в клетку N, M. (путем называется набор индексов проходимых элементов, *кратчайшим путем* - набор индексов тех элементов массива, сумма значений которых минимальна). Двигаться можно только вниз либо вправо.

**Идея решения:**

Допустим, дан *массив* ([рис.8.5](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=2#image.8.5)):



**Рис. 8.5.**

* Предположим, что мы будем двигаться от клетки (1,1) вправо по первой строке. Мы уже выяснили, что в каждую клетку первой строки ведет только один путь (см. "*Арифметический квадрат*"). Тогда пройденный к каждой клетке путь будет вычисляться как сумма значения данного элемента и элемента, стоящего левее ([рис.8.6](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=2#image.8.6)):



**Рис. 8.6.**

* Аналогично заполняем первый столбец ([рис.8.7](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=2#image.8.7)):



**Рис. 8.7.**

* В клетку (2,2) можно попасть двумя путями - из клетки, стоящей выше или клетки, стоящей левее. Анализируем значения этих клеток (клетки, стоящей выше и клетки, стоящей левее). Из той клетки, значение которой меньше мы и придем в (2,2). Значит, значение клетки (2,2) увеличивается на 5 ([рис.8.08](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=2#image.8.08)):



**Рис. 8.08.**

* Аналогично заполняем остальные элементы массива - анализируем содержимое элемента, стоящего сверху и элемента cлева. То значение, которое меньше и прибавляем к содержимому текущего элемента.
* Вычисление *координат*: Движемся в обратном направлении: из клетки (n,m) к клетке (1,1). Анализируем содержимое элемента, стоящего левее и выше текущего. Координаты того элемента, значение которого меньше и выводим на экран ([рис.8.09](http://www.intuit.ru/studies/courses/2293/593/lecture/12759?page=2#image.8.09)):



**Рис. 8.09.**

**Итого:** Кратчайший *путь* - (3,4)-(3,3)-(2,3)-(1,3)-(1,2)-(1,1)

натуральных чисел указывает на направление обхода элементов):



**Рис. 9.1.**

**Идея решения**: Мысленно "разберем" двумерный *массив* на четыре "вложенных" в исходный ([рис.9.2](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=1#image.9.2)). В каждом из них нужно заполнить *ПОБОЧНУЮ ДИАГОНАЛЬ* (опираясь на типовой *алгоритм* обработки квадратного массива относительно диагоналей):



**Задача 2:** Заполнить *массив*, как показано на [табл.9.1](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=1#table.9.1):

|  |
| --- |
| Таблица 9.1. |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 | 2 |
| 1 | 2 | 3 | 3 | 2 | 1 |
| 1 | 2 | 3 | 3 | 2 | 1 |
| 1 | 2 | 2 | 2 | 2 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

**Идея решения:** Разложим исходную матрицу на "вложенные" ([рис.9.3](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=1#image.9.3)):



**Задача 3:** Заполнить *массив*, как показано на [табл.9.2](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=1#table.9.2):

|  |
| --- |
| Таблица 9.2. |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 20 | 21 | 22 | 23 | 24 | 7 |
| 19 | 32 | 33 | 34 | 25 | 8 |
| 18 | 31 | 36 | 35 | 26 | 9 |
| 17 | 30 | 29 | 28 | 27 | 10 |
| 16 | 15 | 14 | 13 | 12 | 11 |

**Дополнительные сведения:** На таком заполнении массива базируется задача "скатерть Улама". В данной задаче квадратный *массив* заполняется по спирали (заполнение происходит не "вовнутрь", как в примере, а "изнутри" массива). Затем "вычеркиваются" все составные числа (см. задачу "*Решето Эратосфена*"). Оставшиеся на своих местах простые числа образуют причудливый узор, названный в честь автора "Скатертью Улама".

**Идея решения:** Разложим исходную матрицу на "вложенные". Каждую из "вложенных" матриц необходимо заполнить по периметру ([рис.9.4](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=1#image.9.4)).



**Задача 4:** Заполните матрицу, как показано на [рис.9.5](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=2#image.9.5):



**Рис. 9.5.**

**Дополнительные сведения**: Есть в информатике классическая задача "**Магический квадрат**" (в "Магическом квадрате" сумма элементов всех строк, всех столбцов и всех диагоналей равна). Решать ее можно различными способами, один из которых называется методом "Террас". Для создания Магического квадрата размерностью n x n (n - нечетное число) необходимо заполнить двумерный *массив* размерностью(2n-1)x(2n-1) так, как в на [рис.9.6](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=2#image.9.6):



**Рис. 9.6.**

Затем "треугольники", выступающие за пределы жирной рамки перенести внутрь таким образом ([табл.9.3](http://www.intuit.ru/studies/courses/2293/593/lecture/12761?page=1#image.9.3)):

|  |
| --- |
| Таблица 9.3. |
| 3 | **16** | 9 | **22** | 15 |
| **20** | 8 | **21** | 14 | **2** |
| 7 | **25** | 13 | **1** | 19 |
| **24** | 12 | **5** | 18 | **6** |
| 11 | **4** | 17 | **10** | 23 |

В полученном двумерном массиве сумма элементов всех строк и всех столбцов равна.

**Идея решения:** Разбиваем исходную матрицу на "вложенные". В каждой матрице заполняем *побочную диагональ*:



**Рис. 9.7.**

**Задача 4:** Определить, пересекаются ли два отрезка, заданные на плоскости координатами своих концов (примеры взаимного расположения отрезков на [рис.10.5](http://www.intuit.ru/studies/courses/2293/593/lecture/12763?page=2#image.10.5)).



**Рис. 10.5.**

**Идею решения** задачи иллюстрирует схема ([рис.10.6](http://www.intuit.ru/studies/courses/2293/593/lecture/12763?page=2#image.10.6)):

1. Отрезки пересекаются если:



**Рис. 10.6.**

1. Отрезки не пересекаются ([рис.10.7](http://www.intuit.ru/studies/courses/2293/593/lecture/12763?page=2#image.10.7)) если …



**Рис. 10.7.**

Следует отметить, что данный метод определения пересечения двух отрезков имеет *исключение*, в случаях:

* если отрезки равной длины и параллельны;
* если отрезки находятся на одной прямой, но не имеют общих точек

**Геометрические задачи**

На плоскости N точек заданы своими координатами. Найти "центральную" точку (точку, сумма расстояний от которой до остальных точек максимальна).

**Идея решения:** Необходимо применить типовой алгоритм формирования групп РАЗМЕЩЕНИЯ БЕЗ ПОВТОРЕНИЙ. Для поиска максимальной суммы расстояний применим типовой алгоритм ПОИСКА МАКСИМАЛЬНОГО ЭЛЕМЕНТА.

На плоскости N точек заданы своими координатами. Найти 2 наиболее удаленные друг от друга точки.

**Идея решения:** Необходимо применить типовой алгоритм формирования групп СОЧЕТАНИЯ БЕЗ ПОВТОРЕНИЙ. Для поиска двух наиболее удаленных точек применим типовой алгоритм ПОИСКА МАКСИМАЛЬНОГО ЭЛЕМЕНТА.

На плоскости N точек заданы своими координатами. Найти минимальный радиус окружности, включающей в себя все точки.

**Идея решения:**

Минимальной будет окружность, на которой находятся хотя бы три точки ([рис.11.2](http://www.intuit.ru/studies/courses/2293/593/lecture/12764?page=2#image.11.2)). Необходимо найти такие три точки, сумма расстояний между которыми максимальна. Необходимо применить типовой алгоритм формирования групп СОЧЕТАНИЯ БЕЗ ПОВТОРЕНИЙ.



**Рис. 11.2.**

Радиус описанной окружности:R=(abc)/(4S)

В приведенных ниже программах находятся координаты трех, наиболее удаленных друг от друга точек. Вычислить R не составит труда (проделайте это самостоятельно).

Упражнения.

* На карте обозначены N станций (названия станций - А1, А2, А3, …). Требуется рассчитать материальные затраты на строительство автомобильных дорог между станциями. Для этого необходимо написать программу, выдающую запрос на ввод с клавиатуры стоимости дороги между двумя всевозможными станциями и выдающую на экран суммарную стоимость затрат на строительство.
* В некоторой фирме составляют график отпусков. Каждый сотрудник уходит 2 раза в год в 2-хнедельный отпуск. Написать программу для вывода на экран табличной ведомости для заполнения графика отпусков с 2 колонками: первой в формате "Месяц-1\_Месяц 2" и второй пустой (в ней сотрудники будут ставить свои ФИО).
* При помощи 5 различных трафаретов нужно нарисовать три картинки в ряд. Выведите на экран всевозможные комбинации.
* На 4 уроках в начальной школе преподают предметы: математику, письмо, чтение, рисование, музыку, физкультуру и труд. Вывести на экран всевозможные варианты расписания предметов на день.

Пусть N=4. Необходимо сформировать различные группы элементов выбираемых из исходного *множества*. Количество элементов в выборке от 1 до N. Элементы исходного *множества* будем хранить в массиве А ([рис. 12.1](http://www.intuit.ru/studies/courses/2293/593/lecture/12766?page=1#image.12.1)):



**Рис. 12.1.**

Составим таблицу, в которой выбранный элемент отметим "1", невыбранный - "0":

|  |
| --- |
| Таблица 12.1. |
| **1** | **2** | **3** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

**Итого**, сформированы группы: {3}, {2}, {2, 3}, {1}, {1,3}, {1, 2}, {1, 2, 3}.

Для формирования групп потребовалось перебрать все варианты комбинаций "0" и "1". Такой метод формирования комбинаторных групп называется "Двоичным перебором", а количество групп будет равно 2n-1.

**Идея решения:** для выбора элементов из исходного *множества* необходимо получить двоичный код (на единицу больше предыдущего). Первый вариант получения нового двоичного кода - перевод счетчика*цикла* i из десятичной в двоичную систему счисления. Второй вариант получения очередного двоичного кода - ищем в массиве двоичных кодов d последний нулевой элемент , заменяем его на единицу и обнуляем все следующие за ним элементы (этот метод называется лексигрофическим порядком).

Количество возможных комбинаций двоичных кодов 2^n-1 (исключаем двоичный код, состоящий из одних нулей).

**Задачи:**

* "Размен монет": дана купюра достоинством X. Требуется разменять ее монетами по 1, 5, 10, 50 рублей.
* Даны гири массами M1, M2, M3, M4. Как можно взвесить предмет массой X?
* Даны N чисел. Выделите из них группы, содержащие от 1 до N элементов, каждая из которых имеет сумму X.

Упражнения.

* Ввести с клавиатуры целое число n. Вывести натуральный ряд двоичных чисел до числа, десятичное представление которого не превосходит n.
* Тур-фирма предлагает разнообразные путевки, хранящиеся в базе данных в виде названий туров и их стоимостей (всего n туров). Сделать выборку из базы данных тех туров, которые подходят покупателю по цене (покупатель рассчитывает приобрести не более трех путевок не менее, чем на m рублей).