

Министерство науки и высшего образования Российской Федерации  
Казанский национальный исследовательский технический университет –  
КАИ им. А.Н. Туполева

Институт компьютерных технологий и защиты информации  
Отделение СПО ИКТЗИ «Колледж информационных технологий»

## **ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ**

Методические указания к лабораторным работам

Казань 2023

Составитель преподаватель СПО ИКТЗИ Мингалиев Заид Зульфатович

Методические указания к лабораторным работам по дисциплине «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ» предназначены для студентов направления подготовки 09.02.07 «Информационные системы и программирование»

**ОГЛАВЛЕНИЕ**

ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ	4
ЛАБОРАТОРНАЯ РАБОТА №4.	5

## **ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ**

По итогам выполнения работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретённые знания и навыки отвечает на пару небольших вопросов преподавателя по составленной программе, возможностям её доработки;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Шаблон оформления отчета представлен в приложении 1. Требования к формированию отчета представлены в приложении 2.

## ЛАБОРАТОРНАЯ РАБОТА №4.

### Программирование алгоритмов преобразования матриц.

#### ЦЕЛЬ РАБОТЫ

Приобрести умения и практические навыки для программирования алгоритмов преобразования матриц.

#### ХОД РАБОТЫ

##### 1) Массивы в C#

**Массив** представляет набор однотипных данных. Объявление массива в C# похоже на объявление переменной за тем исключением, что после указания типа ставятся квадратные скобки:

1	<code>&lt;Тип&gt;[] &lt;название_массива&gt;;</code>
---	--

Например, определим массив целых чисел:

1	<code>int[] numbers;</code>
---	-----------------------------

После определения переменной массива мы можем присвоить ей определенное значение:

1	<code>int[] numbers = new int[5];</code>
---	--

Здесь вначале мы объявили массив `nums`, который будет хранить данные типа `int`. Далее используя операцию `new`, мы выделили память для пяти элементов массива: `new int[4]`. Число 5 еще называется **длиной массива**. При таком определении все элементы получают значение по умолчанию, которое предусмотрено для их типа. Для типа `int` значение по умолчанию - 0.

Также мы сразу можем указать значения для этих элементов:

1	<code>int[] nums2 = new int[4] { 1, 2, 3, 5 };</code>
---	---

2	<code>int[] nums3 = new int[] { 1, 2, 3, 5 };</code>
---	--

3	<code>int[] nums4 = new[] { 1, 2, 3, 5 };</code>
---	--

4	<code>int[] nums5 = { 1, 2, 3, 5 };</code>
---	--

Все перечисленные выше способы будут равноценны.

Для обращения к элементам массива используются **индексы**. Индекс представляет номер элемента в массиве, при этом нумерация начинается с

нуля, поэтому индекс первого элемента будет равен 0. А чтобы обратиться к четвертому элементу в массиве, нам надо использовать индекс 3, к примеру: `nums[3]`.

Используем индексы для получения и установки значений элементов массива:

1	<code>int[] nums = new int[4];</code>
2	<code>nums[0] = 1;</code>
3	<code>nums[1] = 2;</code>
4	<code>nums[2] = 3;</code>
5	<code>nums[3] = 5;</code>
6	<code>Console.WriteLine(nums[3]); // 5</code>

И так как у нас массив определен только для 4 элементов, то мы не можем обратиться, например, к шестому элементу: `nums[5] = 5;`. Если мы так попытаемся сделать, то мы получим исключение **IndexOutOfRangeException**.

## 2) Массивы в C++

Объявление массива в C++ выглядит следующим образом:

1	<code>&lt;Тип&gt; &lt;название_массива&gt;[&lt;кол-во элементов&gt;];</code>
---	--

Например:

1	<code>int numbers[5];</code>
---	------------------------------

Этим кодом мы создали массив типа `int` с именем `numbers` в котором может храниться до 5-ти элементов.

Рассмотрим способ указания значения элементам массива при его инициализации:

1	<code>int arr[] = {0, 1, 2, 3, 4, 5}; // массив будет иметь 6 элементов (от 0 до 5)</code>
2	<code>int mas[100] = {0}; // все 100 элементов будут иметь значение 0</code>

Для того, чтобы задать значения, нужно сразу после объявления массива указать через равно в фигурных скобках требуемые значения.

Для того, чтобы присвоить элементу массива требуемое значение, достаточно указать имя массива и номер требуемого элемента, а затем после знака равенства указать присваиваемое значение:

1	<code>arr[6] = 7 * 4; // arr[6] равен 28</code>
---	---

Для обращения к элементу массива в C++ надо использовать следующую конструкцию:

1	<code>&lt;любой код&gt; &lt;имя массива&gt; [ &lt;№ элемента&gt; ] &lt;любой код&gt;</code>
---	---

Это может выглядеть следующим образом:

1	<code>cout &lt;&lt; arr[10] &lt;&lt; ", " &lt;&lt; arr[11] &lt;&lt; endl;</code>
---	--

Таким образом мы выводим на экран значения одиннадцатого и двенадцатого элементов.

### 3) Многомерные массивы

Массивы характеризуются таким понятием как **ранг** или **количество измерений**. Выше мы рассматривали массивы, которые имеют одно измерение (то есть их ранг равен 1) - такие массивы можно представлять в виде горизонтального ряда элемента. Но массивы также бывают многомерными. У таких массивов количество измерений (то есть ранг) больше 1.

Массивы, которые имеют два измерения (ранг равен 2), называют двухмерными. Например, создадим одномерный и двухмерный массивы, которые имеют одинаковые элементы:

1	<code>int[] nums1 = new int[] { 0, 1, 2, 3, 4, 5 };</code>
2	<code>int[,] nums2 = { { 0, 1, 2 }, { 3, 4, 5 } };</code>

Визуально оба массива можно представить следующим образом:

#### **Одномерный массив nums1**

0	1	2	3	4	5
---	---	---	---	---	---

**Двухмерный массив nums2**

0	1	2
3	4	5

Поскольку массив `nums2` двухмерный, он представляет собой простую таблицу. Все возможные способы определения двухмерных массивов:

1	<code>int[,] nums1;</code>
2	<code>int[,] nums2 = new int[2, 3];</code>
3	<code>int[,] nums3 = new int[2, 3] { { 0, 1, 2 }, { 3, 4, 5 } };</code>
4	<code>int[,] nums4 = new int[,] { { 0, 1, 2 }, { 3, 4, 5 } };</code>
5	<code>int[,] nums5 = new [,]{ { 0, 1, 2 }, { 3, 4, 5 } };</code>
6	<code>int[,] nums6 = { { 0, 1, 2 }, { 3, 4, 5 } };</code>

Массивы могут иметь и большее количество измерений. Объявление трехмерного массива могло бы выглядеть так:

1	<code>int[,,] nums3 = new int[2, 3, 4];</code>
---	--

Соответственно могут быть и четырехмерные массивы, и массивы с большим количеством измерений. Но на практике обычно используются одномерные и двухмерные массивы.

Если мы хотим отдельно пробежаться по каждой строке в таблице, то надо получить количество элементов в размерности. В частности, у каждого массива есть метод `GetUpperBound(dimension)`, который возвращает индекс последнего элемента в определенной размерности. И если мы говорим непосредственно о двухмерном массиве, то первая размерность (с индексом 0), по сути, это и есть таблица. И с помощью выражения `mas.GetUpperBound(0) + 1` можно получить количество строк таблицы,



представленной двумерным массивом. А через `mas.Length / rows` можно получить количество элементов в каждой строке:

1	<code>int[,] mas = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };</code>
2	<code>int rows = mas.GetUpperBound(0) + 1;</code>
3	<code>int columns = mas.Length / rows;</code>
4	<code>// или так</code>
5	<code>// int columns = mas.GetUpperBound(1) + 1;</code>
6	<code>for (int i = 0; i &lt; rows; i++)</code>
7	<code>{</code>
8	<code>    for (int j = 0; j &lt; columns; j++)</code>
9	<code>    {</code>
10	<code>        Console.Write(\$"{mas[i, j]} \t");</code>
11	<code>    }</code>
12	<code>    Console.WriteLine();</code>
13	<code>}</code>

От многомерных массивов надо отличать массив массивов или так называемый «зубчатый массив»:

1	<code>int[][] nums = new int[3][];</code>
2	<code>nums[0] = new int[2] { 1, 2 };</code> <code>// выделяем память для первого подмассива</code>
3	<code>nums[1] = new int[3] { 1, 2, 3 };</code> <code>// выделяем память для второго подмассива</code>
4	<code>nums[2] = new int[5] { 1, 2, 3, 4, 5 };</code> <code>// выделяем память для третьего подмассива</code>

Здесь две группы квадратных скобок указывают, что это массив массивов, то есть такой массив, который в свою очередь содержит в себе другие массивы. Причем длина массива указывается только в первых квадратных скобках, все последующие квадратные скобки должны быть

пусты: `new int[3][[]]`. В данном случае у нас массив `nums` содержит три массива. Причем размерность каждого из этих массивов может не совпадать.

**Зубчатый массив `nums`**

1	2			
1	2	3		
1	2	3	4	5

Примеры массивов представлены на рисунке 2.

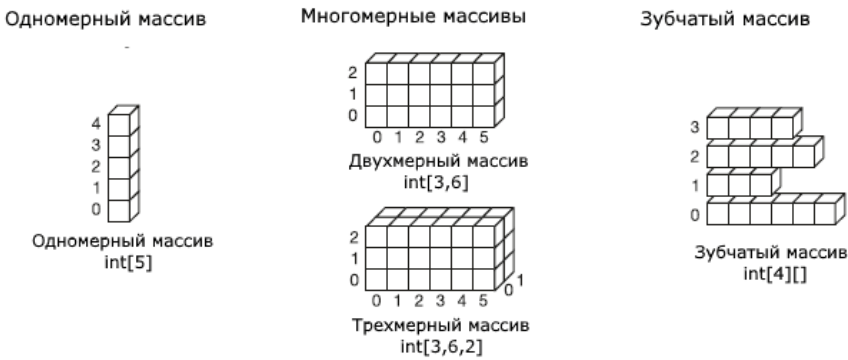


Рисунок 1. Примеры массивов

Причем можно использовать в качестве массивов и многомерные:

1	<code>int[][][,] nums = new int[3][[],]</code>
2	<code>{</code>
3	<code>new int[,] { {1,2}, {3,4} },</code>
4	<code>new int[,] { {1,2}, {3,6} },</code>
5	<code>new int[,] { {1,2}, {3,5}, {8, 13} }</code>
6	<code>};</code>

Так здесь у нас массив из трех массивов, причем каждый из этих массивов представляет двумерный массив.

Используя вложенные циклы, можно перебирать зубчатые массивы.

Например:

1	<code>int[][] numbers = new int[3][];</code>
2	<code>numbers[0] = new int[] { 1, 2 };</code>
3	<code>numbers[1] = new int[] { 1, 2, 3 };</code>
4	<code>numbers[2] = new int[] { 1, 2, 3, 4, 5 };</code>
5	<code>// перебор с помощью цикла for</code>
6	<code>for (int i = 0; i&lt;numbers.Length;i++)</code>
7	<code>{</code>
8	<code>    for (int j =0; j&lt;numbers[i].Length; j++)</code>
9	<code>    {</code>
10	<code>        Console.Write(\$"{numbers[i][j]} \t");</code>
11	<code>    }</code>
12	<code>    Console.WriteLine();</code>
13	<code>}</code>

Суммируем основные понятия массивов:

- Ранг (rank): количество измерений массива
- Длина измерения (dimension length): длина отдельного измерения

массива

- Длина массива (array length): количество всех элементов массива

Например, возьмем массив

1	<code>int[,] numbers = new int[3, 4];</code>
---	--

Массив `numbers` двухмерный, то есть он имеет два измерения, поэтому его ранг равен 2. Длина первого измерения - 3, длина второго измерения - 4. Длина массива (то есть общее количество элементов) - 12.

Для объявления двумерного массива в C++ можно использовать следующую конструкцию:

1	<code>int arr[100][50] = {0}; // двухмерный массив</code>
---	---

Этим кодом мы создали матрицу размером  $100 \times 50$ , которая изначально заполнена нулями. Ее использование ничуть не отличается от выше рассмотренных обычных массивов:

1	<code>int matrix[100][100] = {0};</code>
2	<code>matrix[5][10] = 11;</code>
3	<code>matrix[10][5] = 1;</code>
4	<code>cout &lt;&lt; "Interesting values:\n";</code>
5	<code>cout &lt;&lt; matrix[0][0] &lt;&lt; ", " &lt;&lt; matrix[5][10] &lt;&lt; ", " &lt;&lt; matrix[10][5] &lt;&lt; endl;</code>

Также можно реализовывать трехмерные, четырехмерные и т.д. массивы.

#### 4) Задание на лабораторную работу

Обе части лабораторной работы должны быть выполнены либо на языке программирования C++, либо на языке программирования C#.

#### **Часть 1**

В соответствии с вариантом необходимо написать консольную программу.

Варианты	Индивидуальное задание
1	Ввести 2 массива строк – по 12 элементов в каждом. Сформировать новый массив, на четных местах которого будут элементы с нечетными индексами из первого массива, а на нечетных, с четными индексами из второго.
2	Ввести целочисленный массив, состоящий из 15 элементов. Вычислить сумму цифр всех чисел, входящих в данный массив.
3	Дана действительная квадратная матрица $[a_{ij}]_{i,j=1,n}$ . Размерность матрицы задается пользователем. Получить две квадратные матрицы $[b_{ij}]_{i,j=1,n}$ и $[c_{ij}]_{i,j=1,n}$ , для элементов которых выполняются условия: $b_{ij} = \{a_{ij}, \text{ при } j < i \quad a_{ji}, \text{ при } j \geq i$

Варианты	Индивидуальное задание
	$c_{ij} = \{a_{ij}, \text{ при } j < i - a_{ij}, \text{ при } j \geq i\}$
4	<p>Дана действительная матрица размерности <math>m \times n</math>, в которой не все элементы равны нулю. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.</p>
5	<p>Дана действительная матрица размерности <math>m \times n</math>. Найти сумму наибольшего значений элементов ее строк.</p>
6	<p>В действительной матрице размерности <math>m \times n</math> поменять местами строку, содержащую элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением.</p> <p>Предполагается, что эти элементы единственны.</p>
7	<p>Дана действительная матрица размера <math>m \times n</math>, все элементы в которой различны. В каждой строке выбирается элемент с наибольшим значением, затем среди чисел выбирается наименьшее. Указать индексы элемента с найденным значением.</p>
8	<p>Дана действительная квадратная матрица порядка 12. Заменить нулями все элементы, расположенные на главной диагонали, и единицами, расположенные выше главной диагонали.</p>
9	<p>Дана действительная квадратная матрица порядка 11. Получить целочисленную квадратную матрицу того же порядка, в которой элемент равен единице, если соответствующий ему элемент исходной матрицы больше элемента, расположенного в его строке на главной диагонали, и равен нулю в противоположном случае.</p>
10	<p>Определить, является ли введенная действительная квадратная матрица симметричной относительно своей главной диагонали.</p>

Варианты	Индивидуальное задание																							
11	<p>Известна зарплата каждого из 12 работников фирмы за каждый месяц первого квартала.</p> <table><tr><th rowspan="2">Работники</th><th colspan="3">Месяц</th></tr><tr><th>1</th><th>2</th><th>3</th></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td></tr><tr><td>...</td><td></td><td></td><td></td></tr><tr><td>12</td><td></td><td></td><td></td></tr></table> <p>Организовать ввод информации по этой таблице и определить:</p> <p>а) общую сумму, выплаченную за квартал всем работникам;</p> <p>б) зарплату, полученную за квартал каждым работником;</p> <p>в) общую зарплату всех работников за каждый месяц.</p>	Работники	Месяц			1	2	3	1				2				...				12			
Работники	Месяц																							
	1	2	3																					
1																								
2																								
...																								
12																								
12	<p>Даны натуральные числа <math>i, j</math>, действительная матрица размера <math>18 \times 24</math> (<math>1 \leq i &lt; j \leq 24</math>). Поменять в матрице местами <math>i</math>-й и <math>j</math>-й столбцы.</p>																							
13	<p>Дана действительная квадратная матрица порядка 9. Вычислить сумму тех из ее элементов, расположенных на главной диагонали и выше неё, которые превосходят по величине все элементы, расположенные ниже главной диагонали. Если на главной диагонали и выше неё нет элементов с указанным свойством, то вывести сообщение об этом.</p>																							
14	<p>Будем называть соседями элемента с индексами <math>i, j</math> некоторой матрицы такие элементы этой матрицы, соответствующие индексы которых отличаются от <math>i</math> и <math>j</math> не более чем на единицу.</p> <p>Для данной целочисленной матрицы <math>[a_{ij}]_{i=\overline{1,n}, j=\overline{1,m}}</math> найти матрицу из нулей и единиц <math>[b_{ij}]_{i=\overline{1,n}, j=\overline{1,m}}</math>, элемент которой <math>b_{ij}</math> равен единице, когда все соседи <math>a_{ij}</math> меньше самого <math>a_{ij}</math>.</p>																							

Варианты	Индивидуальное задание
15	Таблица футбольного чемпионата задана квадратной матрицей порядка $n$ , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2 – выигрыш, 1 – ничья, 0 – проигрыш). Найти число команд, имеющих больше побед, чем поражений и определить номера команд, прошедших чемпионат без поражений.
16	Найти решение системы линейных алгебраических уравнений $\{a_{11}x_1 = b_1 \ a_{21}x_1 + a_{22}x_2 = b_2 \ \dots \ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n\}$ . Если задана действительная матрица $[a]_{n \times n}$ и массив свободных членов $[b]_n$ . Для решения СЛАУ можно использовать метод Крамера.
17	В квадратной матрице $n \times n$ обменять значения элементов в каждой строке, расположенные на главной и побочной диагоналях.
18	Сформировать матрицу из чисел от 0 до 999, вывести ее на экран. Посчитать количество двузначных чисел в ней.
19	Записать элементов прямоугольной матрицы в одномерный массив в порядке следования столбцов.
20	Вводятся пять вещественных чисел. Записать в первый столбец матрицы целую часть чисел, во второй - дробную часть, приведенную к пятизначному целому, в третий столбец - знак числа: 0 для положительных чисел и 1 - для отрицательных. Например, если вводится число 5.452456, то в первой ячейке строки присваивается 5, второй присваивается 45245, а третьей - число 0.

## Часть 2.

В соответствии с вариантом необходимо написать консольную программу.

Варианты	Индивидуальное задание
1	<p>Узник пытается бежать из замка, который состоит из <math>MN</math> квадратных комнат, расположенных в виде прямоугольника <math>M \times N</math>. Между любыми двумя соседними комнатами есть дверь, однако некоторые комнаты закрыты и попасть в них нельзя. В начале узник находится в угловой комнате и для спасения ему надо попасть в противоположную угловую комнату. Времени у него немного, всего он может побывать не более, чем в <math>M+N-1</math> комнате, включая начальную и конечную комнату на своем пути, то есть с каждым переходом в соседнюю комнату расстояние до выхода из замка должно уменьшаться. От вас требуется найти количество различных маршрутов, ведущих к спасению.</p> <p>Первая строка входных данных содержит натуральные числа <math>M</math> и <math>N</math>, не превосходящих 1000. Далее идет план замка в виде <math>M</math> строчек из <math>N</math> символов в каждой. Один символ соответствует одной комнате: если символ равен 1, то в комнату можно попасть, если он равен 0, то комната закрыта. Первоначальное положение узника – левый нижний угол (первый символ последней строки), выход находится в правом верхнем углу (последний символ первой строки, оба этих символа равны 1). Программа должна напечатать количество маршрутов, ведущих узника к выходу и проходящих через <math>M+N-1</math> комнату, или слово «Невозможно», если таких маршрутов не существует.</p>
2	<p>Всем хорошо известна "Игра в 15", представляющая собой 15 квадратных фишек, пронумерованных числами от 1 до 15.</p>



Варианты	Индивидуальное задание
	<p>Фишки уложены в квадрат со стороной в 4 стороны фишки, одна позиция для фишки свободна. Если обозначить свободную позицию за *, то головоломка состоит в том, чтобы получить из произвольной начальной позиции позицию следующего вида:</p> <div>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 *</div> <p>Единственной разрешенной операцией является обмен * с одной из соседних по ребру фишек. Операции будем кодировать буквами:</p> <p>R – поменять * с фишкой, которая стоит справа от *</p> <p>L – поменять * с фишкой, которая стоит слева от *</p> <p>U – поменять * с фишкой, которая стоит сверху от *</p> <p>D – поменять * с фишкой, которая стоит снизу от *</p> <p>Например, решением головоломки</p> <div>1 2 3 4 5 6 7 8 9 10 12 * 13 14 11 15</div> <p>является последовательность LDR.</p> <p>От вас требуется решить более простую головоломку "Игра в 8", в которой требуется расположить 8 фишек в виде:</p> <div>1 2 3 4 5 6 7 8 *</div> <p>На вход программе подается описание исходной позиции в виде строки, в которой перечислены номера фишек, разделенные</p>

Варианты	Индивидуальное задание
	<p>пробелами в квадрате слева направо сверху вниз. Вместо свободной фишки напечатана *.</p> <p>Например, позиция</p> <pre> 1  2  3 *  4  6 7  5  8 </pre> <p>задается строкой</p> <pre>1 2 3 * 4 6 7 5 8</pre> <p>На выход программа должна вывести одну строку, состоящую из букв l, r, u, d, содержащую последовательность операций, разрешающую данную головоломку, в которой одна буква соответствует перемещению одной фишки (см. выше правило кодирования операций). Если же головоломка неразрешима, то требуется вывести одно слово unsolvable.</p>
3	<p>На прямоугольном поле для игры в морской бой размером <math>M \times N</math> расположено несколько прямоугольных кораблей. Корабли не соприкасаются друг с другом. Ваша задача — определить всевозможные типы кораблей на поле и число кораблей каждого типа. Два корабля относятся к одному типу, если их размеры совпадают (корабли, которые могут быть получены друг из друга поворотом, также относятся к одному типу).</p> <p>Первая строка входных данных содержит два положительных числа <math>M</math> и <math>N</math>, не превосходящих 1000, задающие размеры поля. Далее идет <math>M</math> строк, каждая из которых состоит из <math>N</math> символов. Символ '1' означает, что соответствующая клетка поля занята кораблем, символ '0' — что свободна. Пробелов в строке нет.</p> <p>Программа должна для каждого обнаруженного типа корабля вывести одну строку, содержащую три числа. Первые два числа</p>

Варианты	Индивидуальное задание
	<p>задают размеры корабля (первое число должно быть не меньше второго), третье число задает количество кораблей данного типа на поле. Строки в выводе должны быть отсортированы по первому числу, затем по второму числу.</p> <p style="text-align: center;"><b>Пример входных данных</b></p> <p style="text-align: center;">6 10</p> <p style="text-align: center;">0111000011</p> <p style="text-align: center;">0000011011</p> <p style="text-align: center;">0100011000</p> <p style="text-align: center;">0101011011</p> <p style="text-align: center;">0100000000</p> <p style="text-align: center;">0001111011</p> <p style="text-align: center;"><b>Пример выходных данных</b></p> <p style="text-align: center;">1 1 1</p> <p style="text-align: center;">2 1 2</p> <p style="text-align: center;">2 2 2</p> <p style="text-align: center;">3 1 2</p> <p style="text-align: center;">3 2 1</p> <p style="text-align: center;">4 1 1</p>
4	<p>На сайте сотового оператора BeepLine сделали защиту от роботов, рассылающих SMS-сообщения: прежде, чем отправить SMS, пользователь должен написать, какую фигуру он видит в специальном окошке: квадрат или круг. Причем, для усиления защиты, в рисунок внесены небольшие помехи.</p> <p>Коле срочно нужно разослать всем друзьям сообщение, поэтому он просит Вас написать программу, распознающую изображение.</p>

Варианты	Индивидуальное задание
	<p>Экспериментально установлено, что система рисует квадрат с помехами следующим образом: сначала на белом фоне рисуется черный квадрат <math>k \times k</math> клеток (<math>k \geq 3</math>), затем некоторые клетки на границе квадрата (на рисунке обозначены цифрой 1) закрашиваются белым, а некоторые клетки (если таковые существуют), граничащие с квадратом (на рисунке обозначены цифрой 2), закрашиваются черным.</p> <div></div> <p>Например, квадрат <math>4 \times 4</math> после нанесения помех может выглядеть так:</p> <div></div> <p>На входе пользователь задает размерность <math>m \times n</math> размера экрана.</p> <p>Следующие <math>n</math> строк, по <math>m</math> символов в каждой, содержат описание картинки. Черные клетки обозначены символом «+», а</p>

Варианты	Индивидуальное задание
	<p>белые - символом «.». Программа на выходе должна определить, какое изображение – квадрат или круг, представлено на изображении. Если изображение не соответствует ни одной из фигур, вывести сообщение об ошибке распознавания.</p>
5	<p>Спецслужбы получили информацию о том, что в труднодоступной части Муравийской пустыни расположена хорошо замаскированная база террористов. В руки спецслужб попал и план этой базы, которая с большой высоты выглядит как группа скальных обломков, весьма часто встречающихся среди Муравийских песков. Президент отдал приказ уничтожить базу крылатыми ракетами. Ваша задача - по карте пустыни, полученной со спутника и плану базы определить количество возможных положений базы террористов.</p> <p>На входе вводятся числа <math>N_b</math> и <math>M_b</math> (<math>1 \leq N_b, M_b \leq 20</math>). В следующих <math>N_b</math> строках записан план базы. Каждая из этих строк содержит по <math>M_b</math> символов "#" (ASCII 35) или "." (ASCII 46). Символ "#" обозначает фрагмент базы, а символ "." - песок.</p> <p>В следующей строке записаны числа <math>N_d</math> и <math>M_d</math> (<math>1 \leq N_d, M_d \leq 100</math>). И остаток входных данных содержит карту участка пустыни, на котором, предположительно, находится база террористов - <math>N_d</math> строк по <math>M_d</math> символов "#" или "." в каждой.</p> <p style="text-align: center;"><b>Пример входных данных:</b></p> <div style="text-align: center;"> <p>2 2</p> <p>#.</p> <p>##</p> <p>3 5</p> </div>

Варианты	Индивидуальное задание
	<p data-bbox="906 253 986 291">#.#.#</p> <p data-bbox="898 318 994 356">#####</p> <p data-bbox="911 383 981 421">.###.</p>