

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

|  |
| --- |
| **институт математики и компьютерных технологий**  **Департамент информационных и компьютерных систем** |

**Отчёт**

по дисциплине **«****Web-дизайн и программирование»**

на тему **«****Программирование»**

|  |  |  |
| --- | --- | --- |
|  |  | Выполнил студент  гр. Б9121-09.03.03пиэ/2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Туровец В. Ю. |
|  |  | Проверил старший преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Шувалова Е. И. |
|  |  |  |

г. Владивосток

2023г.

**Оглавление**

[PHP 3](#_Toc138331085)

[Задание 1 3](#_Toc138331086)

[Задание 2 3](#_Toc138331087)

[Задание 3 4](#_Toc138331088)

[Задание 4 5](#_Toc138331089)

[Задание 5 6](#_Toc138331090)

[Задание 6 6](#_Toc138331091)

[JavaScript 8](#_Toc138331092)

[Задание 1 8](#_Toc138331093)

[Задание 2 9](#_Toc138331094)

[Задание 3 11](#_Toc138331095)

[Python 16](#_Toc138331096)

[Задание 1 16](#_Toc138331097)

[Задание 2 17](#_Toc138331098)

[Задание 3 17](#_Toc138331099)

[Задание 4 18](#_Toc138331100)

[Задание 5 19](#_Toc138331101)

[Задание 6 20](#_Toc138331102)

[Задание 7 21](#_Toc138331103)

[Задание 8 22](#_Toc138331104)

# PHP

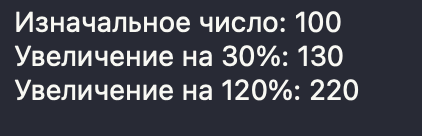
## Задание 1

Дано число. Увеличьте его на 30%, на 120%.

Формализация:

Для увеличения на 30% и на 120% необходимо умножить на 1.3 и 2.2 соответственно, а после вывести на экран результаты.

Вывод программы:



Код программы:

<?php

$number = 100; // Заданное число

echo "Изначальное число: " . $number . "\n";

echo "Увеличение на 30%: " . $number \* 1.3. "\n";

echo "Увеличение на 120%: " . $number \* 2.2. "\n";

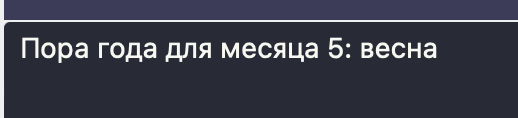
## Задание 2

Дан номер месяца. Вывести название поры года (весна, лето и так далее) или слово 'Ошибка', если месяца с таким номером не существует.

Формализация:

Для упрощения алгоритма лучше всего воспользоваться оператором выбора, который будет сравнивать число и, например, если оно равняется 3, 4 или 5 выводить словно весна, а если число больше 12 или меньше 1 выводить ошибку. В языке php это оператор switch.

Вывод программы:



Код программы:

<?php  
function getSeason($month) {  
 switch ($month) {  
 case 3:  
 case 4:  
 case 5:  
 return "весна";  
 case 6:  
 case 7:  
 case 8:  
 return "лето";  
 case 9:  
 case 10:  
 case 11:  
 return "осень";  
 case 12:  
 case 1:  
 case 2:  
 return "зима";  
 default:  
 return "Ошибка";  
 }  
}  
  
// Пример использования  
$month = 5;  
$season = getSeason($month);  
echo "Пора года для месяца $month: " . $season;

## 

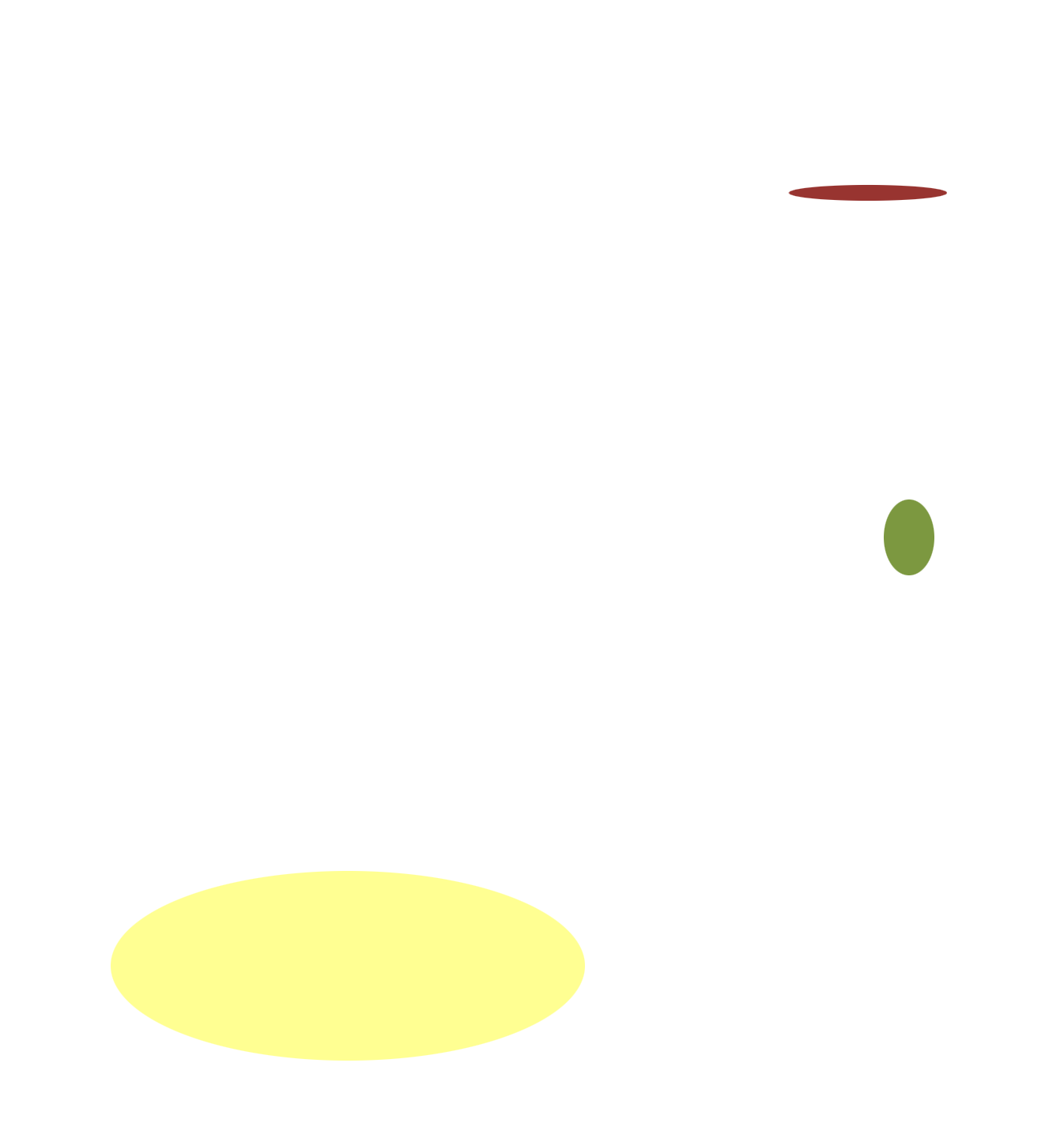
## Задание 3

Пользователь указывает цвет и размер трех кругов. Вывести их в случайных местах в браузере.

Формализация:

Для указания случённого места в браузере необходимо сгенерировать случайные целые числа для координат x и y.

Вывод программы:



Код программы:

<?php  
// Проверяем наличие ключей в массиве $\_POST  
$color1 = isset($\_POST['color1']) ? $\_POST['color1'] : '';  
$size1 = isset($\_POST['size1']) ? $\_POST['size1'] : '';  
  
$color2 = isset($\_POST['color2']) ? $\_POST['color2'] : '';  
$size2 = isset($\_POST['size2']) ? $\_POST['size2'] : '';  
  
$color3 = isset($\_POST['color3']) ? $\_POST['color3'] : '';  
$size3 = isset($\_POST['size3']) ? $\_POST['size3'] : '';  
  
// Генерируем случайные координаты для каждого круга  
$circle1X = rand(0, 800);  
$circle1Y = rand(0, 600);  
  
$circle2X = rand(0, 800);  
$circle2Y = rand(0, 600);  
  
$circle3X = rand(0, 800);  
$circle3Y = rand(0, 600);  
?>  
  
<!DOCTYPE html>  
<html>  
<head>  
 <style>  
 .circle {  
 position: absolute;  
 border-radius: 50%;  
 }  
 </style>  
</head>  
<body>  
<div class="circle" style="background-color: <?php echo $color1; ?>;  
 width: <?php echo $size1; ?>px;  
 height: <?php echo $size1; ?>px;  
 top: <?php echo $circle1Y; ?>px;  
 left: <?php echo $circle1X; ?>px;"></div>  
  
<div class="circle" style="background-color: <?php echo $color2; ?>;  
 width: <?php echo $size2; ?>px;  
 height: <?php echo $size2; ?>px;  
 top: <?php echo $circle2Y; ?>px;  
 left: <?php echo $circle2X; ?>px;"></div>  
  
<div class="circle" style="background-color: <?php echo $color3; ?>;  
 width: <?php echo $size3; ?>px;  
 height: <?php echo $size3; ?>px;  
 top: <?php echo $circle3Y; ?>px;  
 left: <?php echo $circle3X; ?>px;"></div>  
</body>  
</html>

## Задание 4

Дан массив с элементами **1, 2, 3, 4, 5**. С помощью цикла **foreach** найдите **сумму** элементов этого массива. Запишите ее в переменную **$result**.

Формализация:

При помощи цикла с последовательным перебором можно пройтись по массиву, присвоив каждое значение элемента массива итерируемой переменной “element” и прибавлять к другой переменной “result” значение “element”.

Вывод программы:



Код программы:

<?php  
$array = [1, 2, 3, 4, 5];

$result = 0;

echo "Массив: ";

foreach ($array as $element) {

echo $element. " ";

$result += $element;

}

echo "\nСумма элементов массива: " . $result;

## Задание 5

Пользователь вводит названия городов через пробел. Переставьте названия так, чтобы названия были упорядочены по алфавиту.

Формализация:

Для начала необходимо разделить строку на отдельный слова и проанализировать каждое, воспользовавшись методом разделения строки на массив отдельных строк, ориентируясь на пробелы между словами (слово это строка символов слева и справа от пробела) (в php это метод explode). Далее воспользоваться быстрой сортировкой для сортировки массива слов по алфавиту (в php она по умолчанию встроена в метод sort), преобразовать массив в строку, обратно вставив пробелы для лучшего чтения (в php это метод implode) и вывести.

Вывод программы:



Код программы:

<?php  
$input = "Москва Санкт-Петербург Киев Варшава Лондон";  
  
$cityArray = explode(" ", $input);  
  
sort($cityArray);  
  
$sortedCities = implode(" ", $cityArray);  
echo "Города в алфавитном порядке: " . $sortedCities;

## Задание 6

Сделайте функцию **getCommonDivisors**, которая параметром принимает 2 числа, а возвращает массив их общих делителей. Для этого используйте вспомогательную функцию **getDivisors** из предыдущей задачи.

Формализация:

Делить число на все числа меньше или равные ему, после чего проделать тоже самое с другим числом. Сравнить получившиеся два набора делителей и вывести общие для них обоих числа

Вывод программы:



Код программы:

<?php  
function getDivisors($number) {  
 $divisors = [];  
 for ($i = 1; $i <= $number; $i++) {  
 if ($number % $i === 0) {  
 $divisors[] = $i;  
 }  
 }  
 return $divisors;  
}  
  
function getCommonDivisors($number1, $number2) {  
 $divisors1 = getDivisors($number1);  
 $divisors2 = getDivisors($number2);  
 $commonDivisors = array\_intersect($divisors1, $divisors2);  
 return $commonDivisors;  
}  
  
// Пример использования  
$number1 = 36;  
$number2 = 48;  
  
$commonDivisors = getCommonDivisors($number1, $number2);  
  
echo "Общие делители чисел $number1 и $number2: " . implode(", ", $commonDivisors);

# JavaScript

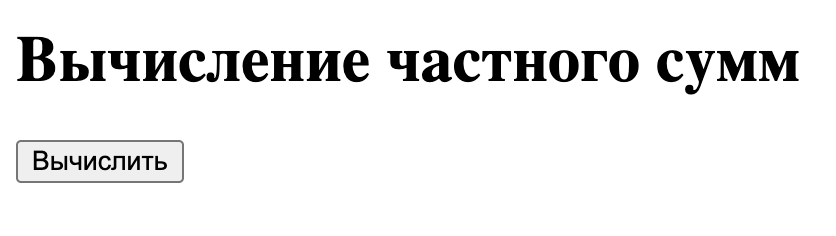
## Задание 1

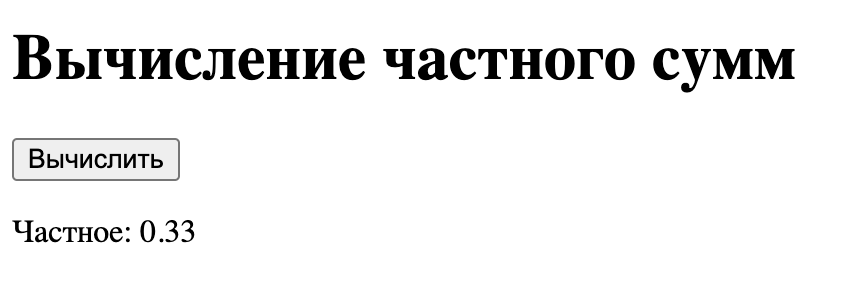
Найти частное (1+3+5 + ...+100)/(101+103+...+200).

Формализация:

Для того чтобы посчитать числитель необходимо циклом от числа 1 до числа 100 пройтись циклом, со всеми этими числами как итерируемыми, складывая их в заранее объявленную переменную, а для знаменателя проделать тоже самое с числами от 101 до 200 и другой заранее объявленной переменной. После чего поделить переменную числителя («numeratorSum») на переменную знаменателя («numeratorSum»).

Вывод программы:





Код программы:

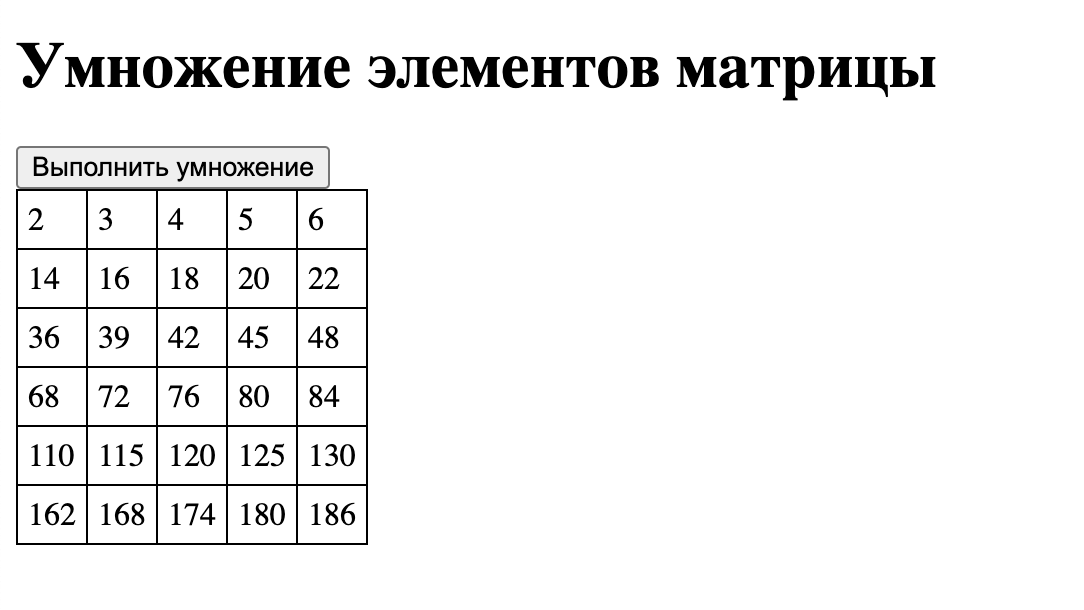
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Вычисление частного сумм</title>  
 <script>  
 function calculateQuotient() {  
 var numeratorSum = 0;  
 var denominatorSum = 0;  
 for (var i = 1; i <= 100; i += 2) {  
 numeratorSum += i;  
 }  
 for (var j = 101; j <= 200; j += 2) {  
 denominatorSum += j;  
 }  
 var quotient = numeratorSum / denominatorSum;  
  
 ***document***.getElementById("result").innerHTML = "Частное: " + quotient.toFixed(2);  
 }  
 </script>  
</head>  
<body>  
<h1>Вычисление частного сумм</h1>  
<button onclick="calculateQuotient()">Вычислить</button>  
<p id="result"></p>  
</body>  
</html>

## Задание 2

Матрица М состоит из 6 строк, 5 столбцов. Умножить эле­менты в каждой из строк на номер этой строки.

Вывод программы:





Код программы:

<!DOCTYPE html>  
<html lang="en">  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Умножение элементов матрицы</title>  
 <style>  
 table {  
 border-collapse: collapse;  
 }  
 th, td {  
 border: 1px solid black;  
 padding: 5px;  
 }  
 </style>  
 <script>  
 function multiplyMatrix() {  
 var matrix = [  
 [2, 3, 4, 5, 6],  
 [7, 8, 9, 10, 11],  
 [12, 13, 14, 15, 16],  
 [17, 18, 19, 20, 21],  
 [22, 23, 24, 25, 26],  
 [27, 28, 29, 30, 31]  
 ];  
 var numRows = matrix.length;  
 for (var i = 0; i < numRows; i++) {  
 var row = matrix[i];  
 var numCols = row.length;  
  
 for (var j = 0; j < numCols; j++) {  
 row[j] = row[j] \* (i + 1);  
 }  
 }  
 displayMatrix(matrix);  
 }  
 function displayMatrix(matrix) {  
 var table = ***document***.getElementById("matrixTable");  
 table.innerHTML = "";  
 var numRows = matrix.length;  
 for (var i = 0; i < numRows; i++) {  
 var row = matrix[i];  
 var numCols = row.length;  
  
 var tableRow = ***document***.createElement("tr");  
  
 for (var j = 0; j < numCols; j++) {  
 var tableData = ***document***.createElement("td");  
 tableData.innerText = row[j];  
 tableRow.appendChild(tableData);  
 }  
 table.appendChild(tableRow);  
 }  
 }  
 </script>  
</head>  
<body>  
<h1>Умножение элементов матрицы</h1>  
<button onclick="multiplyMatrix()">Выполнить умножение</button>  
<table id="matrixTable">  
 <tr>  
 <th>Столбец 1</th>  
 <th>Столбец 2</th>  
 <th>Столбец 3</th>  
 <th>Столбец 4</th>  
 <th>Столбец 5</th>  
 </tr>  
 <tr>  
 <td>2</td>  
 <td>3</td>  
 <td>4</td>  
 <td>5</td>  
 <td>6</td>  
 </tr>  
 <tr>  
 <td>7</td>  
 <td>8</td>  
 <td>9</td>  
 <td>10</td>  
 <td>11</td>  
 </tr>  
 <tr>  
 <td>12</td>  
 <td>13</td>  
 <td>14</td>  
 <td>15</td>  
 <td>16</td>  
 </tr>  
 <tr>  
 <td>17</td>  
 <td>18</td>  
 <td>19</td>  
 <td>20</td>  
 <td>21</td>  
 </tr>  
 <tr>  
 <td>22</td>  
 <td>23</td>  
 <td>24</td>  
 <td>25</td>  
 <td>26</td>  
 </tr>  
 <tr>  
 <td>27</td>  
 <td>28</td>  
 <td>29</td>  
 <td>30</td>  
 <td>31</td>  
 </tr>  
</table>  
</body>  
</html>

## Задание 3

Выполнить задание по вариантам. n - чётное и больше либо равно 4.

При выполнении задания необходимо написать 3 функции:

* Функция getRandomInt(min, max)

Принимает параметры min и max. Возвращает случайное целое число между min (включительно) и max (не включая max)

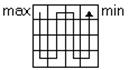
* Функция getArray(n)

Принимает размерность n, возвращает массив со сгенерированными значениями через функцию getRandomInt

* Функция getResultArray(a)

Принимать исходный массив a возвращать отсортированный массив.

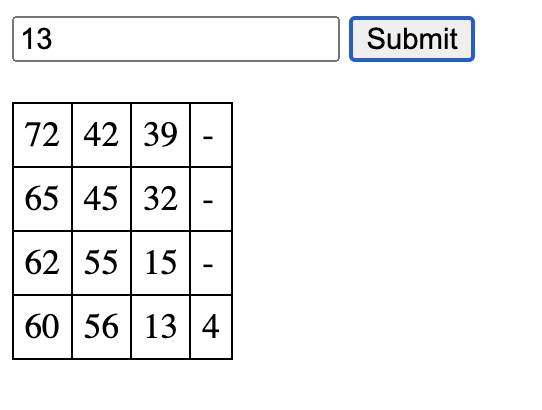
Даны числа a1..an, получить квадратную матрицу порядка n, элементами которой являются числа a1..an, расположенные в ней по схеме:



Формализация:

Для начала необходимо сгенерировать массив случайных чисел длинной в заданное пользователем число, после чего его отсортировать от большего к меньшему. Далее нежно проверить является ли длина массива квадратом какого либо числа и если нет извлекаем из числа квадрат, округляя в верхнюю сторону, и далее заполнить матрицу прочерками до тех пор пока длина массива не будет равна минимально большему квадрату числа (например если число равно 13, то мы должны дополнить массив до 16 элементов) и выводить его матрицей: номер строки является чётным числом, то от большего к меньшему, в случае если номер строки является нечётным - наоборот

Вывод программы:



Код программы:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Вычисление матрицы</title>  
  
 <style>  
 table {  
 border-collapse: collapse;  
 }  
 td {  
 border: 1px solid black;  
 padding: 5px;  
 }  
 </style>  
 <script>  
 function getRandomInt(min, max) {  
 min = ***Math***.ceil(min);  
 max = ***Math***.floor(max);  
 return ***Math***.floor(***Math***.random() \* (max - min)) + min;  
 }  
  
 function getArray(n) {  
 var arr = [];  
 for (var i = 0; i < n; i++) {  
 arr.push(getRandomInt(1, 100)); // Генерируем случайное число от 1 до 100  
 }  
 return arr;  
 }  
 function getResultArray(a) {  
 return a.sort(function(a, b) {  
 return b - a;  
 });  
 }  
  
 function isSquareRootable(number) {  
 if (number < 0) {  
 return false; // Отрицательные числа не имеют корня  
 }  
  
 var squareRoot = ***Math***.sqrt(number);  
 return squareRoot % 1 === 0; // Проверяем, является ли квадратный корень целым числом  
 }  
  
 function makeSquareArray(numbers) {  
 var squareSize = ***Math***.ceil(***Math***.sqrt(numbers.length)); // Определяем размер квадрата (округленный вверх до ближайшего целого)  
 var missingNumbers = squareSize \* squareSize - numbers.length; // Вычисляем количество недостающих чисел  
 var zerosArray = ***Array***(missingNumbers).fill("-"); // Создаем массив из нулей для заполнения  
  
 return numbers.concat(zerosArray); // Объединяем исходный массив с массивом из нулей  
 }  
  
  
 function generateMatrix(n, numbers) {  
 var size = ***Math***.sqrt(n);  
 var matrix = [];  
 var index = 0;  
  
  
 // Создаем пустую квадратную матрицу размером size x size  
 for (var i = 0; i < size; i++) {  
 matrix[i] = [];  
 }  
  
 // Заполняем матрицу элементами в указанной схеме  
 for (var col = 0; col < size; col++) {  
 if (col % 2 === 0) {  
 // Заполняем столбцы с нечетными индексами в прямом порядке  
 for (var row = 0; row < size; row++) {  
 matrix[row][col] = numbers[index];  
 index++;  
 }  
 } else {  
 // Заполняем столбцы с четными индексами в обратном порядке  
 for (var row = size - 1; row >= 0; row--) {  
 matrix[row][col] = numbers[index];  
 index++;  
 }  
 }  
 }  
  
 return matrix;  
 }  
  
 function generateMatrixHTML(n, numbers) {  
 var size = ***Math***.sqrt(n);  
 var matrix = generateMatrix(n, numbers);  
  
 var html = '<table>';  
 for (var i = 0; i < size; i++) {  
 html += '<tr>';  
 for (var j = 0; j < size; j++) {  
 html += '<td>' + matrix[i][j] + '</td>';  
 }  
 html += '</tr>';  
 }  
 html += '</table>';  
  
 return html;  
 }  
  
  
 function handleInput() {  
 var numberInput = ***document***.getElementById('numberInput');  
 var inputValue = parseInt(numberInput.value);  
  
 n = inputValue  
  
 var numbers = getArray(n); // Получаем исходный массив  
 ***console***.log("Исходный массив:", numbers);  
  
 var sortedArray = getResultArray(numbers); // Получаем отсортированный массив  
 ***console***.log("Отсортированный массив:", sortedArray);  
  
 if (!isSquareRootable(n)){  
 numbers = makeSquareArray(numbers)  
 }  
  
 n = numbers.length  
  
 var matrixHTML = generateMatrixHTML(n, numbers);  
 ***document***.getElementById('matrix-container').innerHTML = matrixHTML;  
 }  
  
 </script>  
</head>  
<body>  
 <input type="number" id="numberInput">  
 <button onclick="handleInput()">Submit</button>  
 <div id="output"></div>  
 <br>  
 <div id="matrix-container"></div>  
</body>  
</html>

# Python

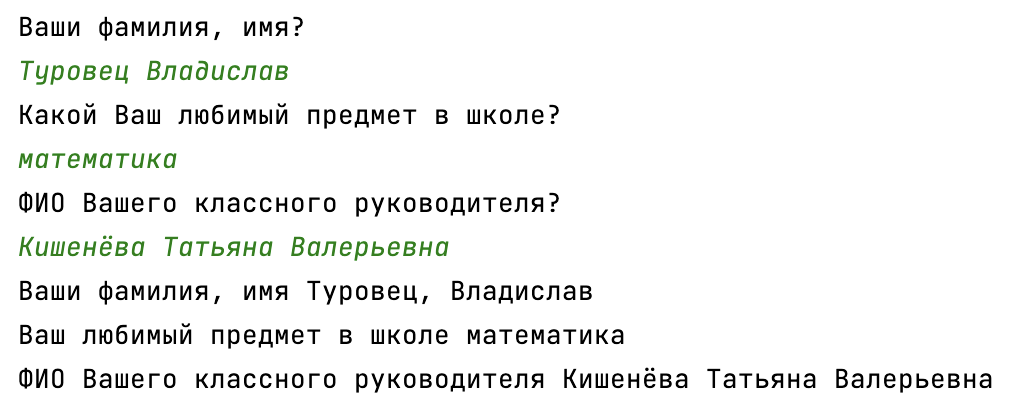
## Задание 1

Напишите программу, которая запрашивала бы у пользователя:

Имя, Фамилия, Любимый предмет в школе (в институте), ФИО классного руководителя (куратора)

- Фамилия, имя ( "Ваши фамилия, имя?")  
- любимый предмет в школе ("Какой Ваш любимый предмет в школе?")  
- ФИО классного руководителя ("ФИО Вашего классного руководителя?") После этого выводила бы три строки:  
"Ваши имя, фамилия"  
"Ваш любимый предмет в школе "  
"ФИО Вашего классного руководителя"

Вывод программы:

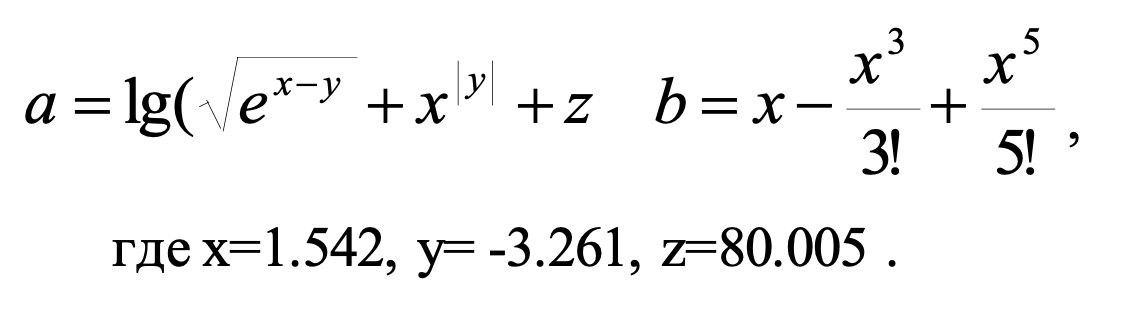


Код программы:

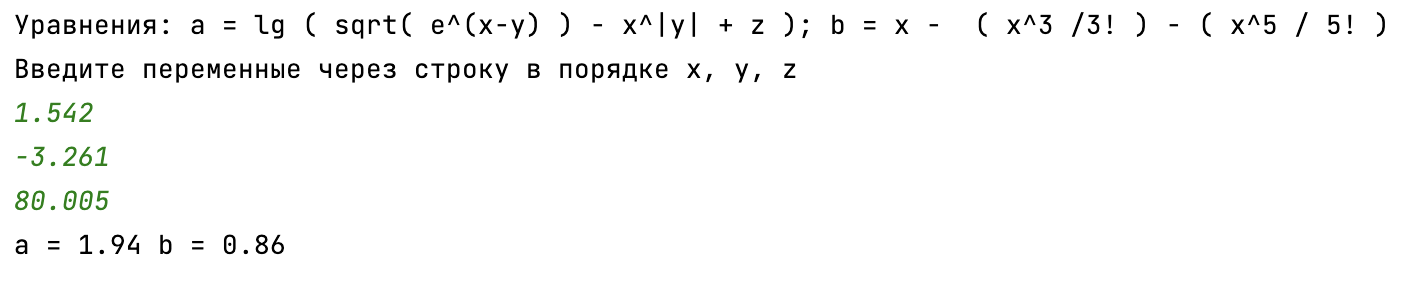
Line = input(**"Ваши фамилия, имя?\n"**).split(**" "**)  
Name, Surname = Line[0], Line[1]  
Subject = input(**"Какой Ваш любимый предмет в школе?\n"**)  
Teacher = input(**"ФИО Вашего классного руководителя?\n"**)  
  
print(**f"Ваши фамилия, имя {**Name**}, {**Surname**}"**)  
print(**f"Ваш любимый предмет в школе {**Subject**}"**)  
print(**f"ФИО Вашего классного руководителя {**Teacher**}"**)

## Задание 2

Напишите программу, которая бы вычисляла заданное арифметическое выражение при заданных переменных. Ввод переменных осуществляется с клавиатуры. Вывести результат с 2-мя знаками после запятой.



Вывод программы:



Код программы:

**from** math **import** \*  
print(**"Уравнения: a = lg ( sqrt( e^(x-y) ) - x^|y| + z ); b = x - ( x^3 /3! ) - ( x^5 / 5! )"**)  
print(**"Введите переменные через строку в порядке x, y, z"**)  
x = float(input())  
y = float(input())  
z = float(input())  
  
a = log10( sqrt( e\*\*(x-y) ) - x\*\*fabs(y) + z )  
b = x - ( x\*\*3 / perm(3) ) - ( x\*\*5 / perm(5) )  
  
print(**f"a = {**a**:.2f} b = {**b**:.2f}"**)

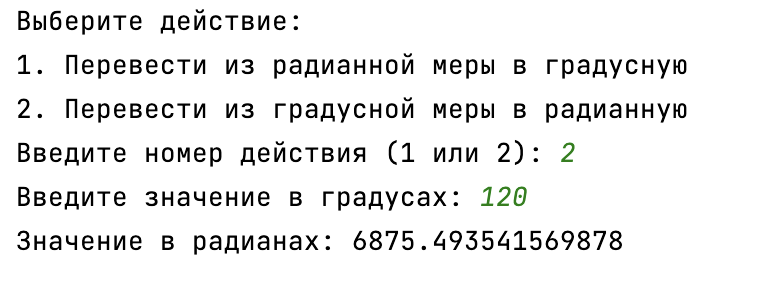
## Задание 3

Составить программу, осуществляющую перевод величин из радианной меры в градусную или наоборот. Программа должна запрашивать, какой перевод нужно осуществить, и выполнять указанное действие.

Формализация:

Для того чтобы перевести градусы в радианы необходимо умножить число градусов на π и разделить на 180, а для того, чтобы перевести радианы в градусы необходимо разделить число градусов на π и умножить на 180. (Для удобства в библиотеке math есть уже готовые функции перевода radians и degrees)

Вывод программы:



Код программы:

**from** math **import** \*

print(**"Выберите действие:"**)  
print(**"1. Перевести из радианной меры в градусную"**)  
print(**"2. Перевести из градусной меры в радианную"**)  
choice = int(input(**"Введите номер действия (1 или 2): "**))  
  
**if** choice == 1:  
 rad = float(input(**"Введите значение в радианах: "**))  
 deg = radians(rad)  
 print(**f"Значение в градусах: {**deg**}"**)  
**elif** choice == 2:  
 deg = float(input(**"Введите значение в градусах: "**))  
 rad = degrees(deg)  
 print(**f"Значение в радианах: {**rad**}"**)  
**else**:  
 print(**"Некорректный выбор действия."**)

## Задание 4

Известны оценки по физике каждого из 20 учеников класса. Определить среднюю оценку. Решить задачу используя циклическую конструкцию for.

Формализация:

Средняя оценка — это сумма всех оценок, делённая на количество учеников (в нашем случае 20)

Вывод программы:



Код программы:

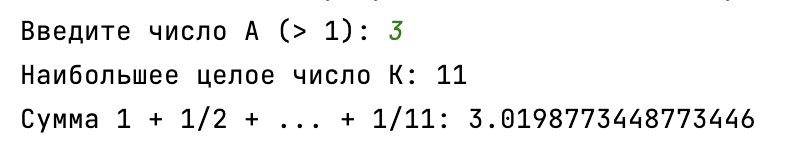
num\_students = 20  
total\_grades = 0  
  
**for** i **in** range(1, num\_students + 1):  
 grade = float(input(**f"Введите оценку для ученика {**i**}: "**))  
 total\_grades += grade  
  
average\_grade = total\_grades / num\_students  
print(**f"Средняя оценка по физике: {**average\_grade**}"**)

2. Дано число A (> 1). Вывести наибольшее из целых чисел K, для которых сумма 1 + 1/2 + ... + 1/K будет меньше A, и саму эту сумму. Решить задачу используя циклическую конструкцию while.

Формализация:

Спрашиваем у пользователя значение числа А, после чего создаём переменную суммы и в цикле складываем итерируемую переменную K от 1 до тех пор, пока переменная суммы меньше A.

Вывод программы:



Код программы:

A = int(input(**"Введите число A (> 1): "**))  
K = 1  
current\_sum = 0  
  
**while** current\_sum < A:  
 current\_sum += 1 / K  
 K += 1  
  
K -= 1 *# Уменьшаем K на 1, так как последнее значение K уже превысило A*print(**f"Наибольшее целое число K: {**K**}"**)  
print(**f"Сумма 1 + 1/2 + ... + 1/{**K**}: {**current\_sum**}"**)

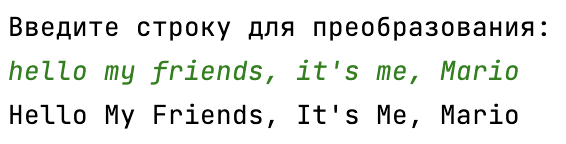
## Задание 5

Дана строка-предложение на английском языке. Преобразовать строку так, чтобы каждое слово начиналось с заглавной буквы.

Формализация:

Разделить строку на массив слова, ориентируясь по пробелам) и далее каждый первый элемент каждого слова массива привести к верхнему регистру.

Вывод программы:



Код программы:

words = input(**"Введите строку для преобразования: \n"**).split(**" "**)  
upperWords = **""  
for** word **in** words:  
 upperWords += word[0].upper() + word[1:] + **" "**print(upperWords)

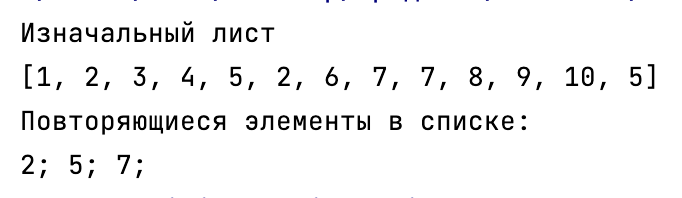
## Задание 6

1. Определите, есть ли в списке повторяющиеся элементы, если да, то вывести на экран это значение, иначе сообщение об их отсутствии.

Формализация:

Пройтись по каждому элементу листа, если число есть в массиве уникальных элементов добавить его в массив дубликатов, если нет – в массив уникальных элементов.

Вывод программы:



Код программы:

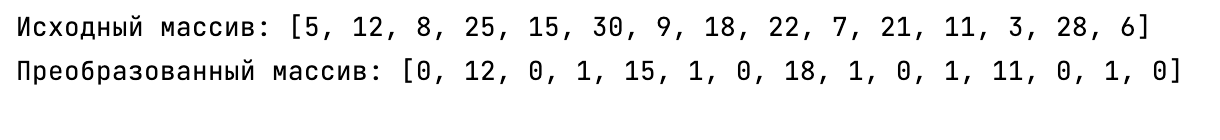
my\_list = [1, 2, 3, 4, 5, 2, 6, 7, 7, 8, 9, 10, 5]  
print(**f"Изначальный лист \n{**my\_list**}"**)  
  
duplicates = set()  
unique\_elements = set()  
  
**for** element **in** my\_list:  
 **if** element **in** unique\_elements:  
 duplicates.add(element)  
 **else**:  
 unique\_elements.add(element)  
  
**if** duplicates:  
 print(**"Повторяющиеся элементы в списке:"**)  
 **for** element **in** duplicates:  
 print(**f"{**element**}"**, end=**"; "**)  
**else**:  
 print(**"Повторяющихся элементов в списке нет."**)

2. Дан одномерный массив из 15 элементов. Элементам массива меньше 10 присвоить нулевые значения, а элементам больше 20 присвоить 1. Вывести на экран монитора первоначальный и преобразованный массивы в строчку.

Формализация:

Необходимо пройтись по массиву с помощью цикла и после если число меньше 10 присвоить ему значение 1, а если больше 20 - значение 1.

Вывод программы:



Код программы:

array = [5, 12, 8, 25, 15, 30, 9, 18, 22, 7, 21, 11, 3, 28, 6]  
  
transformed\_array = []  
**for** element **in** array:  
 **if** element < 10:  
 transformed\_array.append(0)  
 **elif** element > 20:  
 transformed\_array.append(1)  
 **else**:  
 transformed\_array.append(element)  
  
print(**"Исходный массив:"**, array)  
print(**"Преобразованный массив:"**, transformed\_array)

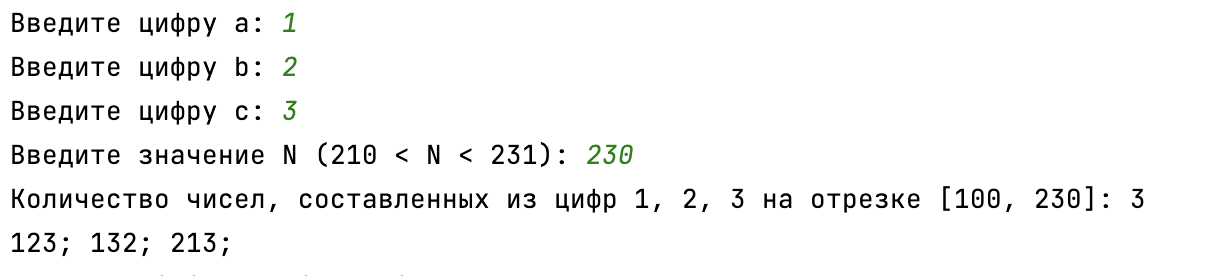
## Задание 7

1. На отрезке [100, N] (210 < N < 231) найти количество чисел, составленных из цифр а, b, с.

Формализация:

Для начала считываем цифры a, b, c и N (где 210 < N < 231), после чего создаём переменную-счётчик и пустой массив для записи всех подходящих под условие чисел, проходимся по всему отрезку чисел с помощью цикла, если число содержит в себе все три цифры a, b, c мы увеличиваем счётчик на 1 и добавляем цифру в массив. В конце мы выводим значение счётчика и все подходящие числа.

Вывод программы:



Код программы:

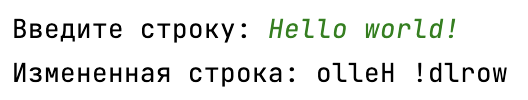
**def** count\_numbers(a, b, c, N):  
 count = 0  
 result = []  
  
 **for** num **in** range(100, N + 1):  
 digits = str(num)  
 **if** (str(a) **in** digits) & (str(b) **in** digits) & (str(c) **in** digits):  
 result.append(digits)  
 count += 1  
  
  
 **return** count, result  
  
  
a = int(input(**"Введите цифру а: "**))  
b = int(input(**"Введите цифру b: "**))  
c = int(input(**"Введите цифру c: "**))  
N = int(input(**"Введите значение N (210 < N < 231): "**))  
  
**if** 210 < N < 231:  
 count, result = count\_numbers(a, b, c, N)  
 print(**f"Количество чисел, составленных из цифр {**a**}, {**b**}, {**c**} на отрезке [100, {**N**}]: {**count**}"**)  
 **for** number **in** result: print(number, end=**"; "**)  
**else**:  
 print(**"Некорректное значение N."**)

2. Составить программу, которая изменяет последовательность слов в строке на обратную.

Формализация:

Для начала считываем всю строку, после чего чего разделяем её на массив слов. Далее проходимся по каждому слову в массиве слов и записываем слова в обратном порядке в новый массив слов. После чего записываем слова в строку через пробел.

Вывод программы:



Код программы:

**def** reverse\_string(sentence):  
 words = sentence.split()  
 newWords = **""  
 for** word **in** words:  
 newWords += word[::-1] + **" "** reversed\_sentence = **''**.join(newWords)  
 **return** reversed\_sentence  
  
  
sentence = input(**"Введите строку: "**)  
reversed\_sentence = reverse\_string(sentence)  
print(**"Измененная строка:"**, reversed\_sentence)

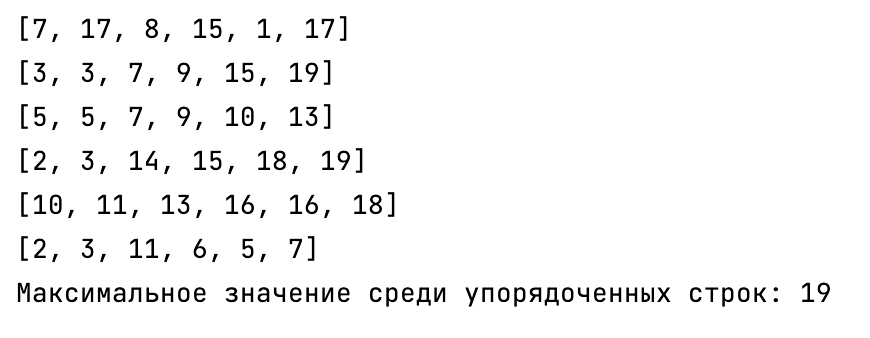
## Задание 8

1. Найти максимальный среди всех элементов тех строк заданной матрицы, которые упорядочены (либо по возрастанию, либо по убыванию).

Формализация:

Инициализировать переменную для хранения максимального значения. Проверить, является ли строка упорядоченной. Найти масимальное значение в упорядоченно строке, если максимальное значение не является числом или меньше максимального значения в упорядоченной строке. Сгенерировать матрицу 6х6 со случайными цифрами, выбрать случайные 4 строки для сортировки. Проверить нужно ли сортировать данную строку, если да – отсортировать.

Вывод программы:

****

Код программы:

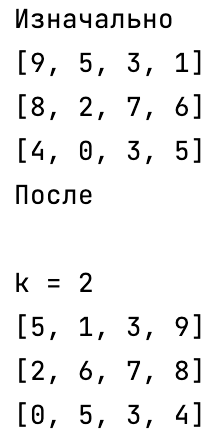
**import** random  
  
**def** find\_max\_ordered(matrix):  
 max\_value = **None** *# Инициализируем переменную для хранения максимального значения* **for** row **in** matrix:  
 **if** sorted(row) == row **or** sorted(row, reverse=**True**) == row: *# Проверяем, является ли строка упорядоченной* row\_max = max(row) *# Находим максимальное значение в упорядоченной строке* **if** max\_value **is None or** row\_max > max\_value: *# Обновляем максимальное значение, если требуется* max\_value = row\_max  
 **return** max\_value  
  
  
**def** generate\_and\_sort():  
 matrix = [[random.randint(1, 20) **for** \_ **in** range(6)] **for** \_ **in** range(6)] *# Генерируем матрицу 6x6 со случайными цифрами* rows\_to\_sort = random.sample(range(6), 4) *# Выбираем случайные 4 строки для сортировки* sorted\_matrix = []  
  
 **for** i, row **in** enumerate(matrix):  
 **if** i **in** rows\_to\_sort: *# Проверяем, нужно ли сортировать данную строку* sorted\_row = sorted(row) *# Сортируем строку* sorted\_matrix.append(sorted\_row)  
 **else**:  
 sorted\_matrix.append(row)  
  
 **return** sorted\_matrix  
  
matrix = generate\_and\_sort()  
  
**for** row **in** matrix:  
 print(row)  
  
max\_ordered\_value = find\_max\_ordered(matrix)  
print(**"Максимальное значение среди упорядоченных строк:"**, max\_ordered\_value)

2. Расположить столбцы матрицы D[M, N] в порядке возрастания элементов k-й строки (1 <= k <= М).

Формализация:

Записать количество столбцов матрицы в переменную «num\_colomns», получить k-ю строку матрицы. Создать список индексов столбцов, отсортировать индексы столбцов на основе значений k-й строки. Создать новую матрицу с отсортированными столбцами.

Вывод программы:



Код программы:

**def** sort\_columns\_by\_row(matrix, k):  
 *# Получаем количество столбцов матрицы* num\_columns = len(matrix[0])  
  
 *# Получаем k-ю строку матрицы* row = matrix[k - 1]  
  
 *# Создаем список индексов столбцов* column\_indices = list(range(num\_columns))  
  
 *# Сортируем индексы столбцов на основе значений k-й строки* sorted\_columns = sorted(column\_indices, key=**lambda** i: row[i])  
  
 *# Создаем новую матрицу с отсортированными столбцами* sorted\_matrix = [[row[i] **for** i **in** sorted\_columns] **for** row **in** matrix]  
  
 **return** sorted\_matrix  
  
*# Пример использования*D = [  
 [9, 5, 3, 1],  
 [8, 2, 7, 6],  
 [4, 0, 3, 5]  
]  
  
print(**f"Изначально"**)  
**for** row **in** D:  
 print(row)  
  
  
M = len(D)  
N = len(D[0])  
k = 2  
  
print(**f"После"**)  
print(**f"\nk = {**k**}"**)  
  
sorted\_D = sort\_columns\_by\_row(D, k)  
**for** row **in** sorted\_D:  
 print(row)