

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий
Кафедра информатики, вычислительной техники
и информационной безопасности

Отчет защищен с оценкой _____
Преподаватель _____ *С. В. Умбетов*
« _____ » _____ 2023 г.

Отчёт по лабораторной работе №5
по дисциплине «Алгоритмизация и
программирование»
«Двумерные массивы»

ЛР 09.03.03.5.005

Студент группы ПИЭ-21 _____ *А. Ю. Гончаров*
группа и.о., фамилия

Преподаватель ассистент, к. т. н. _____ *С. В. Умбетов*
должность, ученая степень и.о., фамилия

Лабораторная работа №5

Двумерные массивы

Цели и задачи работы: Изучение алгоритмов формирования и обработки двумерных массивов, программирование и отладка программ формирования и обработки двумерных массивов.

Задание к работе: Реализовать программирование и отладку программ формирования и обработки двумерных массивов. Самостоятельно решить задачи в соответствии с индивидуальным вариантом.

Задание к выполнению принял: _____



Гончаров А.Ю.

Подпись

ФИО

Ход работы

Задание №1. Даны положительные числа M , N , число D и набор из M чисел. Сформировать матрицу размера $M \times N$, у которой первый столбец совпадает с исходным набором чисел, а элементы каждого следующего столбца равны сумме соответствующего элемента предыдущего столбца и числа D (в результате каждая строка матрицы будет содержать элементы арифметической прогрессии).

Составим блок-схему (рисунок 1) для решения данной задачи, которая представлена в варианте №5.

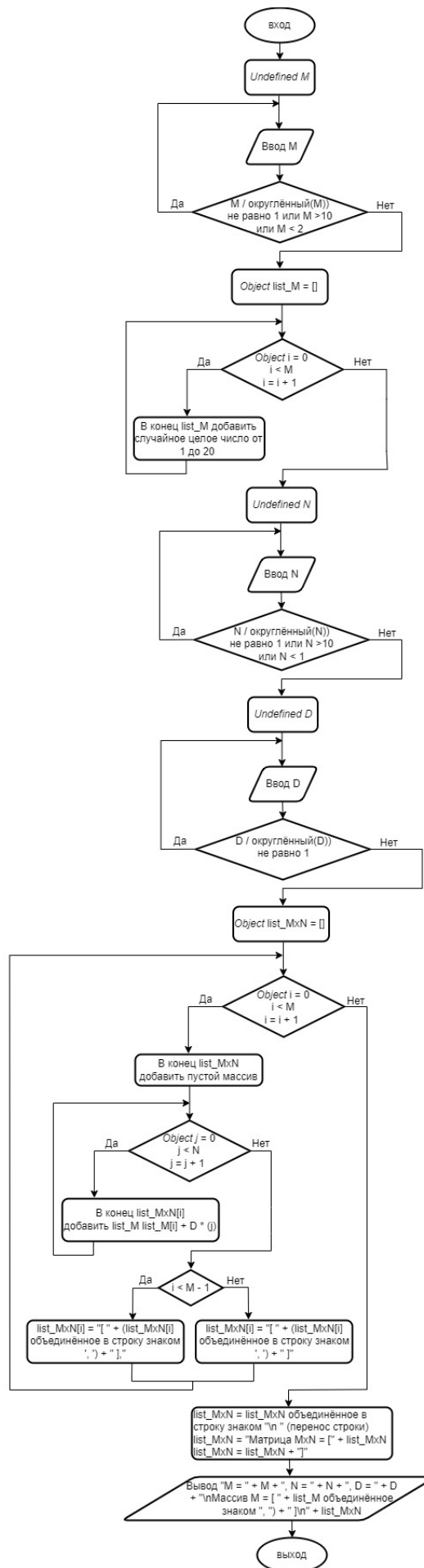
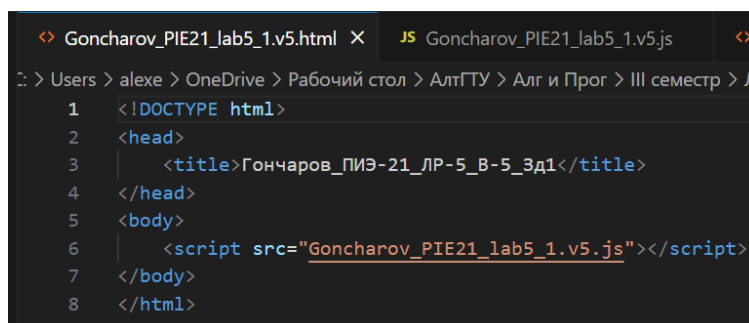


Рисунок 1 - блок-схема для задания №1

Напишем код программы в среде Microsoft Visual Studio Code. Программа представляет из себя два связанных файла. Первый файл (рисунок 2) расширения .html содержит гипертекстовую разметку Web-страницы (html-код), в том числе тег «script», содержащий ссылку на второй файл. Второй файл (рисунки 3-4) расширения .js содержит привязанный к html-странице JavaScript-код, реализующий ввод данных, формирование и обработку двумерных массивов в циклических процессах и вывод результатов.

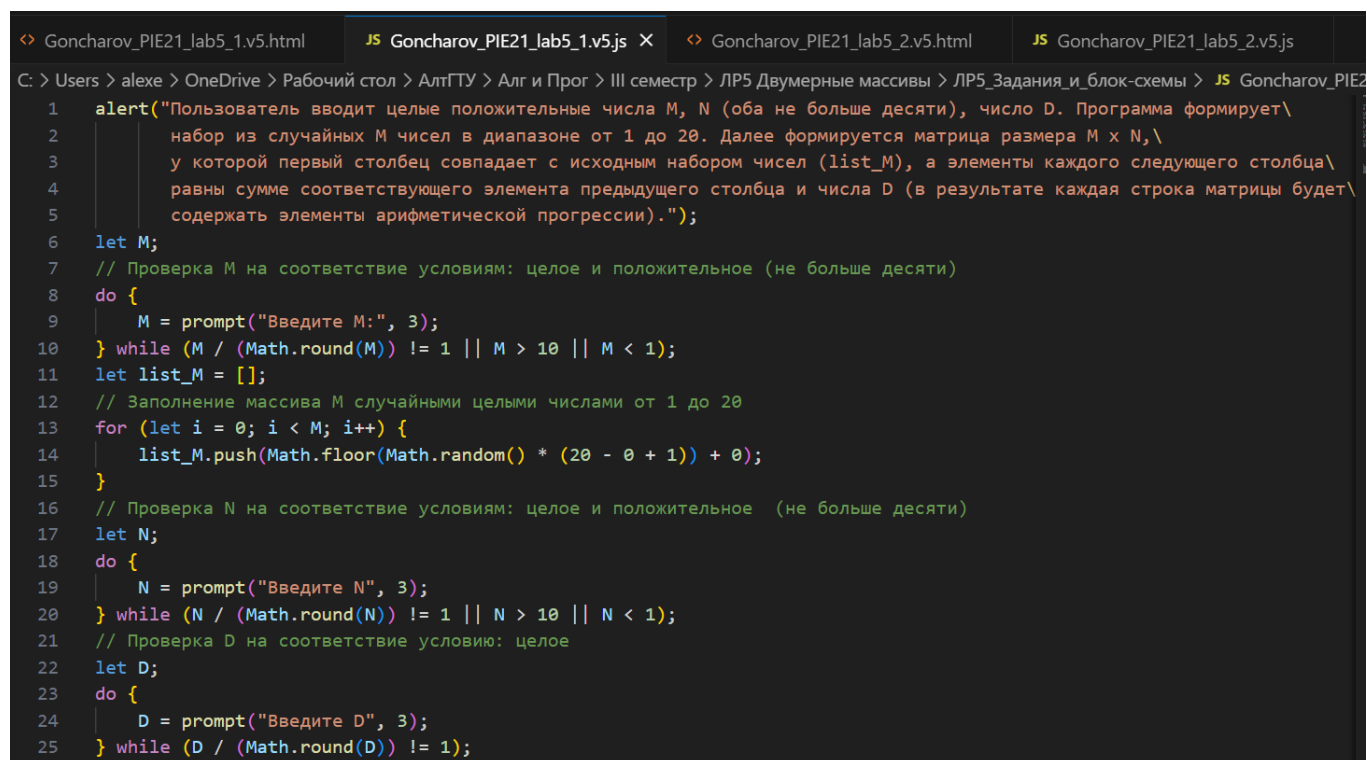


```

1 <!DOCTYPE html>
2 <head>
3   <title>Гончаров_ПИА-21_ЛР-5_В-5_Зд1</title>
4 </head>
5 <body>
6   <script src="Goncharov_PIE21_lab5_1.v5.js"></script>
7 </body>
8 </html>

```

Рисунок 2 – гипертекстовая разметка Web-страницы программы по заданию №1



```

1 alert("Пользователь вводит целые положительные числа M, N (оба не больше десяти), число D. Программа формирует\
2   набор из случайных M чисел в диапазоне от 1 до 20. Далее формируется матрица размера M x N,\
3   у которой первый столбец совпадает с исходным набором чисел (list_M), а элементы каждого следующего столбца\
4   равны сумме соответствующего элемента предыдущего столбца и числа D (в результате каждая строка матрицы будет\
5   содержать элементы арифметической прогрессии).");
6 let M;
7 // Проверка M на соответствие условиям: целое и положительное (не больше десяти)
8 do {
9   M = prompt("Введите M:", 3);
10 } while (M / (Math.round(M)) != 1 || M > 10 || M < 1);
11 let list_M = [];
12 // Заполнение массива M случайными целыми числами от 1 до 20
13 for (let i = 0; i < M; i++) {
14   list_M.push(Math.floor(Math.random() * (20 - 0 + 1)) + 0);
15 }
16 // Проверка N на соответствие условиям: целое и положительное (не больше десяти)
17 let N;
18 do {
19   N = prompt("Введите N", 3);
20 } while (N / (Math.round(N)) != 1 || N > 10 || N < 1);
21 // Проверка D на соответствие условию: целое
22 let D;
23 do {
24   D = prompt("Введите D", 3);
25 } while (D / (Math.round(D)) != 1);

```

Рисунок 3 – привязанный к html-странице JavaScript-код по заданию №1 (первая часть)

```
<> Goncharov_PIE21_lab5_1.v5.html JS Goncharov_PIE21_lab5_1.v5.js X <> Goncharov_PIE21_lab5_2.v5.html JS Goncharov_PIE21_lab5_2.v5.js
C: > Users > alexe > OneDrive > Рабочий стол > АлтГТУ > Алг и Прог > III семестр > ЛР5 Двумерные массивы > ЛР5_Задания_и_блок-схемы > JS Gonchar
21 // Проверка D на соответствие условию: целое
22 let D;
23 do {
24     D = prompt("Введите D", 3);
25 } while (D / (Math.round(D)) != 1);
26 // Заполнение двумерного массива MxN числами в соответствии с условием
27 let list_MxN = [];
28 for (let i = 0; i < M; i++) {
29     list_MxN.push([]);
30     for (let j = 0; j < N; j++) {
31         list_MxN[i].push(list_M[i] + D * (j));
32     }
33     if (i < M - 1) {
34         list_MxN[i] = "[ " + list_MxN[i].join(', ') + " ],";
35     } else {
36         list_MxN[i] = "[ " + list_MxN[i].join(', ') + " ]";
37     }
38 }
39 list_MxN = list_MxN.join("\n");
40 list_MxN = "Матрица MxN = [" + list_MxN;
41 list_MxN = list_MxN + "]\n";
42 alert("M = " + M + ", N = " + N + ", D = " + D + "\nМассив M = [ " + list_M.join(", ") + " ]\n" + list_MxN);
```

Рисунок 4 – привязанный к html-странице JavaScript-код по заданию №1 (вторая часть)

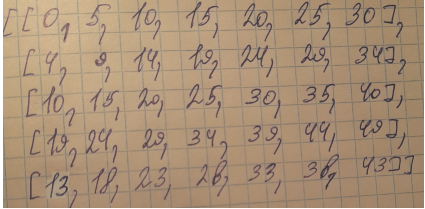
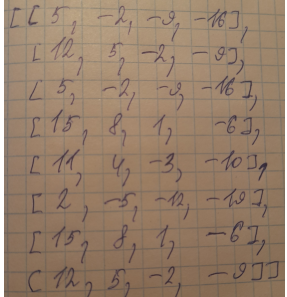
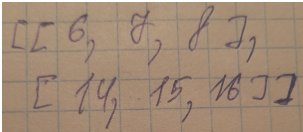
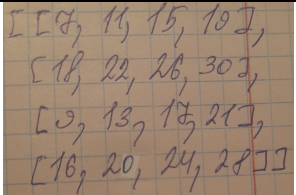
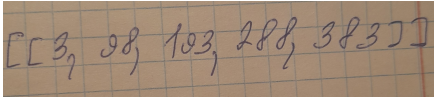
В написанной программе реализована проверка вводимого значения на соответствие условиям задания на всех этапах. Например, программа запрашивает значение N, пока оно не начнёт соответствовать следующим условиям: N - натуральное число от 1 до 10 включительно. После ввода корректного значения происходит переход к следующим инструкциям. Пример “реакции” программы показан на рисунках 5-6.

Рисунок 5 – ввод строки вместо натурального числа от 1 до 10 включительно

Рисунок 6 – повторный запрос программой необходимой величины

Проведём тестирование написанной программы и проверку работы Visual Studio Code ручным способом на тетрадном листке с помощью преобразований в уме, поскольку математическая сложность данных действий совсем небольшая, а реализация проверки в электронных таблицах излишне трудоёмка для такого задания. Ниже в таблице 1 представлено сравнение результатов работы проверяемой программы и проверочные результаты.

Таблица 1 – Сравнение результатов выполнения задания №1

	Номер проверки и исходные данные	Результаты работы программы (матрица MxN)	Проверочные результаты (матрица MxN)	Свер-ка
Исходные данные и новые матрицы	№1 (M=5, N=7, D=5, массив M = [0, 4, 10, 19, 13])	[[0, 5, 10, 15, 20, 25, 30], [4, 9, 14, 19, 24, 29, 34], [10, 15, 20, 25, 30, 35, 40], [19, 24, 29, 34, 39, 44, 49], [13, 18, 23, 28, 33, 38, 43]]		Да
	№2 (M=8, N=4, D=-7, массив M = [5, 12, 5, 15, 11, 2, 15, 12])	[[5, -2, -9, -16], [12, 5, -2, -9], [5, -2, -9, -16], [15, 8, 1, -6], [11, 4, -3, -10], [2, -5, -12, -19], [15, 8, 1, -6], [12, 5, -2, -9]]		Да
	№3 (M=2, N=3, D=1, массив M = [6, 14])	[[6, 7, 8], [14, 15, 16]]		Да
	№4 (M=4, N=4, D=4, массив M = [7, 18, 9, 16])	[[7, 11, 15, 19], [18, 22, 26, 30], [9, 13, 17, 21], [16, 20, 24, 28]]		Да
	№5 (M=1, N=5, D=95, массив M = [0, 4, 10, 19, 13])	[[3, 98, 193, 288, 383]]		Да

Задание №2. Дана матрица размера $M \times N$ (M - чётное число). Поменять местами верхнюю и нижнюю половины матрицы.

Составим блок-схему (рисунок 7) для решения данной задачи, которая представлена в варианте №5.

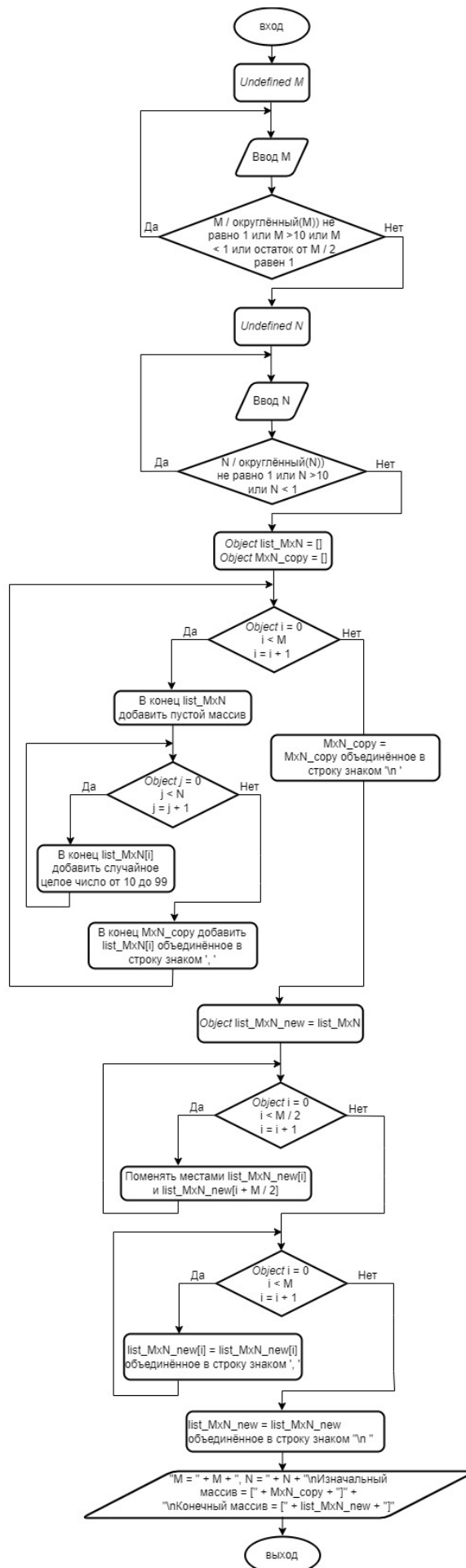
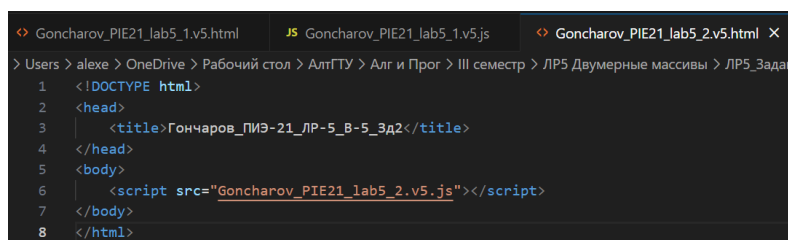


Рисунок 7 - блок-схема для задания №2

Напишем код программы в среде Microsoft Visual Studio Code. Программа представляет из себя два связанных файла. Первый файл (рисунок 8) расширения .html содержит гипертекстовую разметку Web-страницы (html-код), в том числе тег «script», содержащий ссылку на второй файл. Второй файл (рисунки 9-10) расширения .js содержит привязанный к html-странице JavaScript-код, реализующий ввод данных, формирование и обработку двумерных массивов в циклических процессах и вывод результатов.

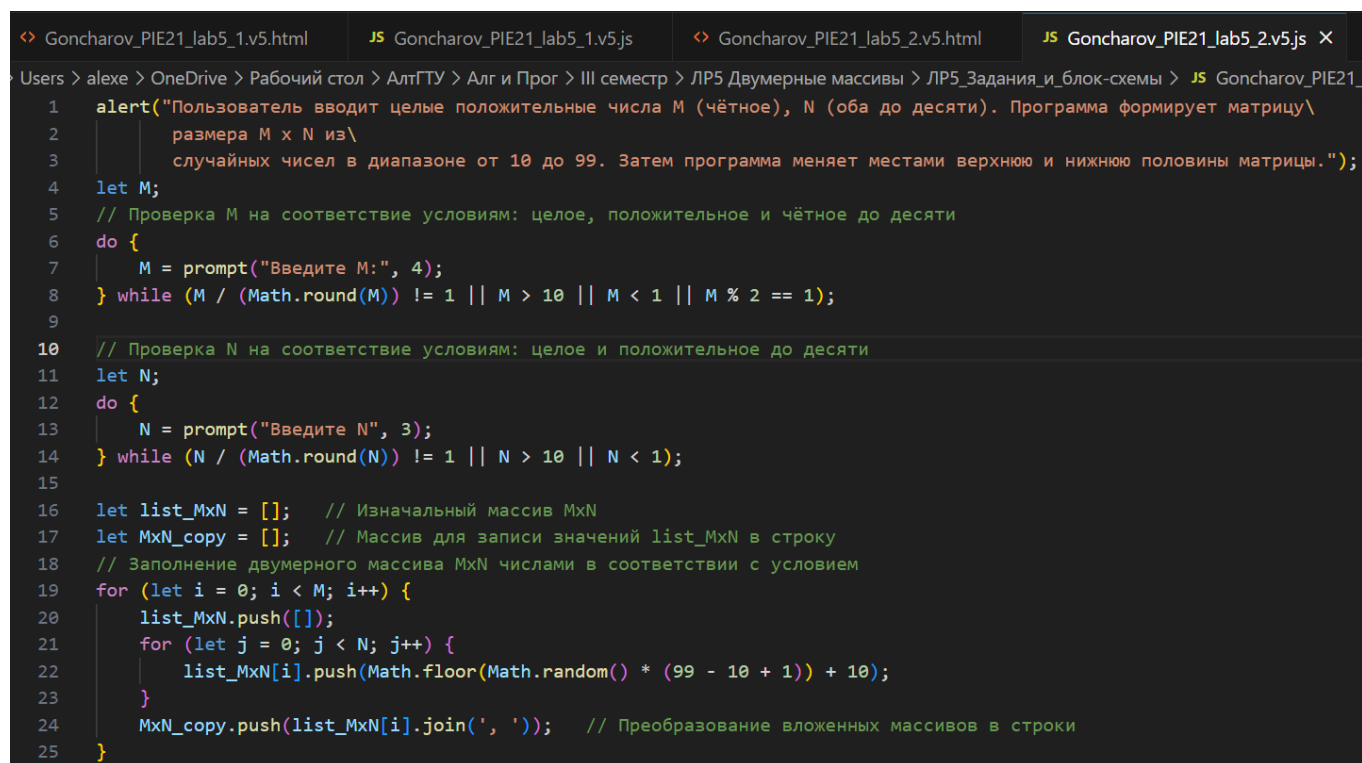


```

1 <!DOCTYPE html>
2 <head>
3   <title>Гончаров_ПИА-21_ЛР-5_В-5_3д2</title>
4 </head>
5 <body>
6   <script src="Goncharov_PIE21_lab5_2.v5.js"></script>
7 </body>
8 </html>

```

Рисунок 8 – гипертекстовая разметка Web-страницы программы по заданию №2



```

1 alert("Пользователь вводит целые положительные числа М (чётное), N (оба до десяти). Программа формирует матрицу\
2   размера М x N из\
3   случайных чисел в диапазоне от 10 до 99. Затем программа меняет местами верхнюю и нижнюю половины матрицы.");
4 let M;
5 // Проверка M на соответствие условиям: целое, положительное и чётное до десяти
6 do {
7   M = prompt("Введите M:", 4);
8 } while (M / (Math.round(M)) != 1 || M > 10 || M < 1 || M % 2 == 1);
9
10 // Проверка N на соответствие условиям: целое и положительное до десяти
11 let N;
12 do {
13   N = prompt("Введите N", 3);
14 } while (N / (Math.round(N)) != 1 || N > 10 || N < 1);
15
16 let list_MxN = []; // Изначальный массив MxN
17 let MxN_copy = []; // Массив для записи значений list_MxN в строку
18 // Заполнение двумерного массива MxN числами в соответствии с условием
19 for (let i = 0; i < M; i++) {
20   list_MxN.push([]);
21   for (let j = 0; j < N; j++) {
22     list_MxN[i].push(Math.floor(Math.random() * (99 - 10 + 1)) + 10);
23   }
24   MxN_copy.push(list_MxN[i].join(' ')); // Преобразование вложенных массивов в строки
25 }

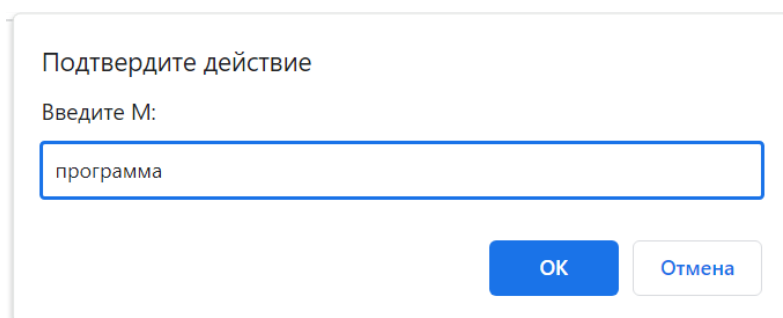
```

Рисунок 9 – привязанный к html-странице JavaScript-код по заданию №2 (вторая часть)

```
<> Goncharov_PIE21_lab5_1.v5.html JS Goncharov_PIE21_lab5_1.v5.js <> Goncharov_PIE21_lab5_2.v5.html JS Goncharov_PIE21_lab5_2.v5.js X
> Users > alexe > OneDrive > Рабочий стол > АлгГТУ > Алг и Прог > III семестр > ЛР5 Двумерные массивы > ЛР5_Задания и блок-схемы > JS Goncharov_PIE21_lab5_2.v5.js
17 let MxN_copy = []; // Массив для записи значений list_MxN в строку
18 // Заполнение двумерного массива MxN числами в соответствии с условием
19 for (let i = 0; i < M; i++) {
20     list_MxN.push([]);
21     for (let j = 0; j < N; j++) {
22         list_MxN[i].push(Math.floor(Math.random() * (99 - 10 + 1)) + 10);
23     }
24     MxN_copy.push(list_MxN[i].join(' ')); // Преобразование вложенных массивов в строки
25 }
26 MxN_copy = MxN_copy.join("\n"); // Разбиение на строки для вывода матрицы
27 // Перестановка верхней и нижней половин матрицы
28 let list_MxN_new = list_MxN;
29 for (let i = 0; i < M / 2; i++) {
30     [list_MxN_new[i], list_MxN_new[i + M / 2]] = [list_MxN_new[i + M / 2], list_MxN_new[i]];
31 }
32 // Преобразование вложенных массивов в строки
33 for (let i = 0; i < M; i++) {
34     list_MxN_new[i] = list_MxN_new[i].join(' ');
35 }
36 list_MxN_new = list_MxN_new.join("\n"); // Разбиение на строки для вывода матрицы
37 alert("M = " + M + ", N = " + N + "\nИзначальный массив = [" + MxN_copy + "] +
38     "\nКонечный массив = [" + list_MxN_new + "]);
```

Рисунок 10 – привязанный к html-странице JavaScript-код по заданию №2 (вторая часть)

В написанной программе реализована проверка вводимого значения на соответствие условиям задания на всех этапах. Например, программа запрашивает значение М, пока оно не начнёт соответствовать следующим условиям: М - целое положительное чётное число не больше десяти. После ввода корректного значения происходит переход к следующим инструкциям. Пример “реакции” программы показан на рисунках 11-12.



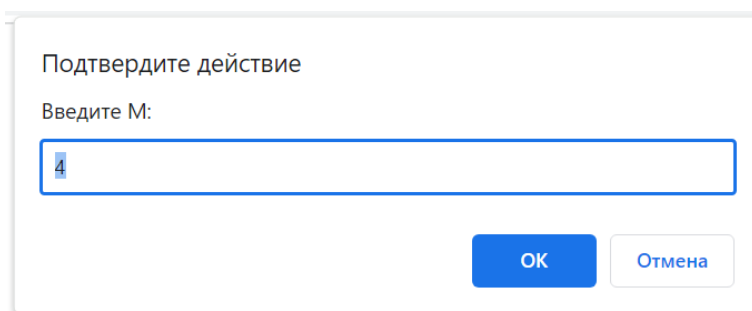
Подтвердите действие

Введите М:

программа

ОК Отмена

Рисунок 11 – ввод текста вместо целого числа



Подтвердите действие

Введите М:

4

ОК Отмена

Рисунок 12 – повторный запрос программой необходимой величины

Проведём тестирование написанной программы и проверку работы Visual Studio Code ручным способом на тетрадном листке с помощью преобразований в уме, поскольку логическая сложность данных действий совсем небольшая, а реализация проверки в электронных таблицах излишне трудоёмка для такого задания. Ниже в таблице 2 представлено сравнение результатов работы проверяемой программы и проверочные результаты.

Таблица 2 – Сравнение результатов выполнения задания №2

	Номер проверки и исходные данные	Результаты работы программы (матрица MxN)	Проверочные результаты (матрица MxN)	Свер- ка																															
Исходные данные и новые матрицы	№1 (M=4, N=3, [61, 10, 39 82, 19, 12 65, 86, 39 81, 68, 81])	[65, 86, 39 81, 68, 81 61, 10, 39 82, 19, 12]	<table><tr><td>65</td><td>86</td><td>39</td></tr><tr><td>81</td><td>68</td><td>81</td></tr><tr><td>61</td><td>10</td><td>39</td></tr><tr><td>82</td><td>19</td><td>12</td></tr></table>	65	86	39	81	68	81	61	10	39	82	19	12	Да																			
	65	86	39																																
	81	68	81																																
	61	10	39																																
	82	19	12																																
№2 (M=8, N=4, [41, 37, 68, 14 41, 48, 95, 75 73, 44, 11, 78 12, 76, 57, 59 66, 42, 28, 55 29, 78, 61, 92 44, 13, 50, 96 83, 41, 80, 86])	[66, 42, 28, 55 29, 78, 61, 92 44, 13, 50, 96 83, 41, 80, 86 41, 37, 68, 14 41, 48, 95, 75 73, 44, 11, 78 12, 76, 57, 59]	<table><tr><td>66</td><td>42</td><td>28</td><td>55</td></tr><tr><td>29</td><td>78</td><td>61</td><td>92</td></tr><tr><td>44</td><td>13</td><td>50</td><td>96</td></tr><tr><td>83</td><td>41</td><td>80</td><td>86</td></tr><tr><td>41</td><td>37</td><td>68</td><td>14</td></tr><tr><td>41</td><td>48</td><td>95</td><td>75</td></tr><tr><td>73</td><td>44</td><td>11</td><td>78</td></tr><tr><td>12</td><td>76</td><td>57</td><td>59</td></tr></table>	66	42	28	55	29	78	61	92	44	13	50	96	83	41	80	86	41	37	68	14	41	48	95	75	73	44	11	78	12	76	57	59	Да
66	42	28	55																																
29	78	61	92																																
44	13	50	96																																
83	41	80	86																																
41	37	68	14																																
41	48	95	75																																
73	44	11	78																																
12	76	57	59																																
№3 (M=2, N=5, [74, 29, 53, 10, 50 47, 71, 57, 20, 22])	[47, 71, 57, 20, 22 74, 29, 53, 10, 50]	<table><tr><td>47</td><td>71</td><td>57</td><td>20</td><td>22</td></tr><tr><td>74</td><td>29</td><td>53</td><td>10</td><td>50</td></tr></table>	47	71	57	20	22	74	29	53	10	50	Да																						
47	71	57	20	22																															
74	29	53	10	50																															
№4 (M=6, N=5, [85, 36, 69, 33, 11 80, 33, 55, 66, 54 78, 71, 40, 13, 99 68, 97, 60, 49, 86 70, 51, 13, 81, 28 16, 69, 34, 35, 59])	[68, 97, 60, 49, 86 70, 51, 13, 81, 28 16, 69, 34, 35, 59 85, 36, 69, 33, 11 80, 33, 55, 66, 54 78, 71, 40, 13, 99]	<table><tr><td>68</td><td>97</td><td>60</td><td>49</td><td>86</td></tr><tr><td>70</td><td>51</td><td>13</td><td>81</td><td>28</td></tr><tr><td>16</td><td>69</td><td>34</td><td>35</td><td>59</td></tr><tr><td>85</td><td>36</td><td>69</td><td>33</td><td>11</td></tr><tr><td>80</td><td>33</td><td>55</td><td>66</td><td>54</td></tr><tr><td>78</td><td>71</td><td>40</td><td>13</td><td>99</td></tr></table>	68	97	60	49	86	70	51	13	81	28	16	69	34	35	59	85	36	69	33	11	80	33	55	66	54	78	71	40	13	99	Да		
68	97	60	49	86																															
70	51	13	81	28																															
16	69	34	35	59																															
85	36	69	33	11																															
80	33	55	66	54																															
78	71	40	13	99																															
№5 (M=2, N=10, [89, 76, 68, 61, 56, 51, 58, 29, 43, 18 88, 86, 17, 42, 52, 98, 31, 38, 89, 52])	[88, 86, 17, 42, 52, 98, 31, 38, 89, 52 89, 76, 68, 61, 56, 51, 58, 29, 43, 18]	<table><tr><td>88</td><td>86</td><td>17</td><td>42</td><td>52</td><td>98</td><td>31</td><td>38</td><td>89</td><td>52</td></tr><tr><td>89</td><td>76</td><td>68</td><td>61</td><td>56</td><td>51</td><td>58</td><td>29</td><td>43</td><td>18</td></tr></table>	88	86	17	42	52	98	31	38	89	52	89	76	68	61	56	51	58	29	43	18	Да												
88	86	17	42	52	98	31	38	89	52																										
89	76	68	61	56	51	58	29	43	18																										

Вывод

В ходе выполнения лабораторной работы я изучил алгоритмы формирования и обработки двумерных массивов, ознакомился с особенностями таких массивов в языке JavaScript посредством программирования и отладки циклических программ формирования и обработки двумерных массивов.

В написанных программах для реализации “прохода” по матрице мной были использованы вложенные циклы типа “for”.

В ходе выполнения работы особенного внимания потребовало изучение механизмов преобразования вложенных массивов в строки.