

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*
« _____ » _____ 2024 г.

Отчёт по лабораторной работе №10
по дисциплине «Разработка кода
информационных систем»
ЛР 09.03.01.14.002

Студент группы 1ИСП-22 _____ *С.М. Шубкин*
группа и.о., фамилия

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

Лабораторная работа №10

Двумерные массивы.

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

Задание к работе:

Задание 1. Дана матрица размера $M \times N$. Для каждой строки матрицы с нечетным номером (1, 3, ...) найти среднее арифметическое ее элементов. Условный оператор не использовать.

Задание 2. Дана квадратная матрица A порядка M . Найти максимальный элемент для каждой ее диагонали, параллельной побочной (начиная с одноэлементной диагонали $A_{1,1}$).



Задание принял: _____

Подпись

Шубкин С.М.

ФИО

Ход работы

Задание №1. Дана матрица размера $M \times N$. Для каждой строки матрицы с нечетным номером (1, 3, ...) найти среднее арифметическое ее элементов. Условный оператор не использовать.

Текст программы:

HTML ▼

```
1 <button id="btnCalc" onclick="calculate()">
2   Calc
3 </button>
4
5 <div id="divResult"> </div>
```

JavaScript + Без библиотеки (чистый JS) ▼

```
1 function calculate(matrix) {
2   const sredznach = [];
3   for (let i = 0; i < matrix.length; i += 2) {
4     let sum = 0;
5     for (let j = 0; j < matrix[i].length; j++) {
6       sum += matrix[i][j];
7     }
8     sredznach.push(sum / matrix[i].length);
9   }
10  return sredznach;
11 }
12
13
14 const matrix = [
15   [1, 2, 3],
16   [4, 5, 6],
17   [7, 8, 9],
18   [10, 11, 12],
19   [14, 45, 78],
20   [45, 67, 113]
21 ];
22
23 const rez = calculate(matrix);
24 document.getElementById("divResult").innerHTML = rez;
25
```

Рисунок 1 – Код программы

Блок-схема:

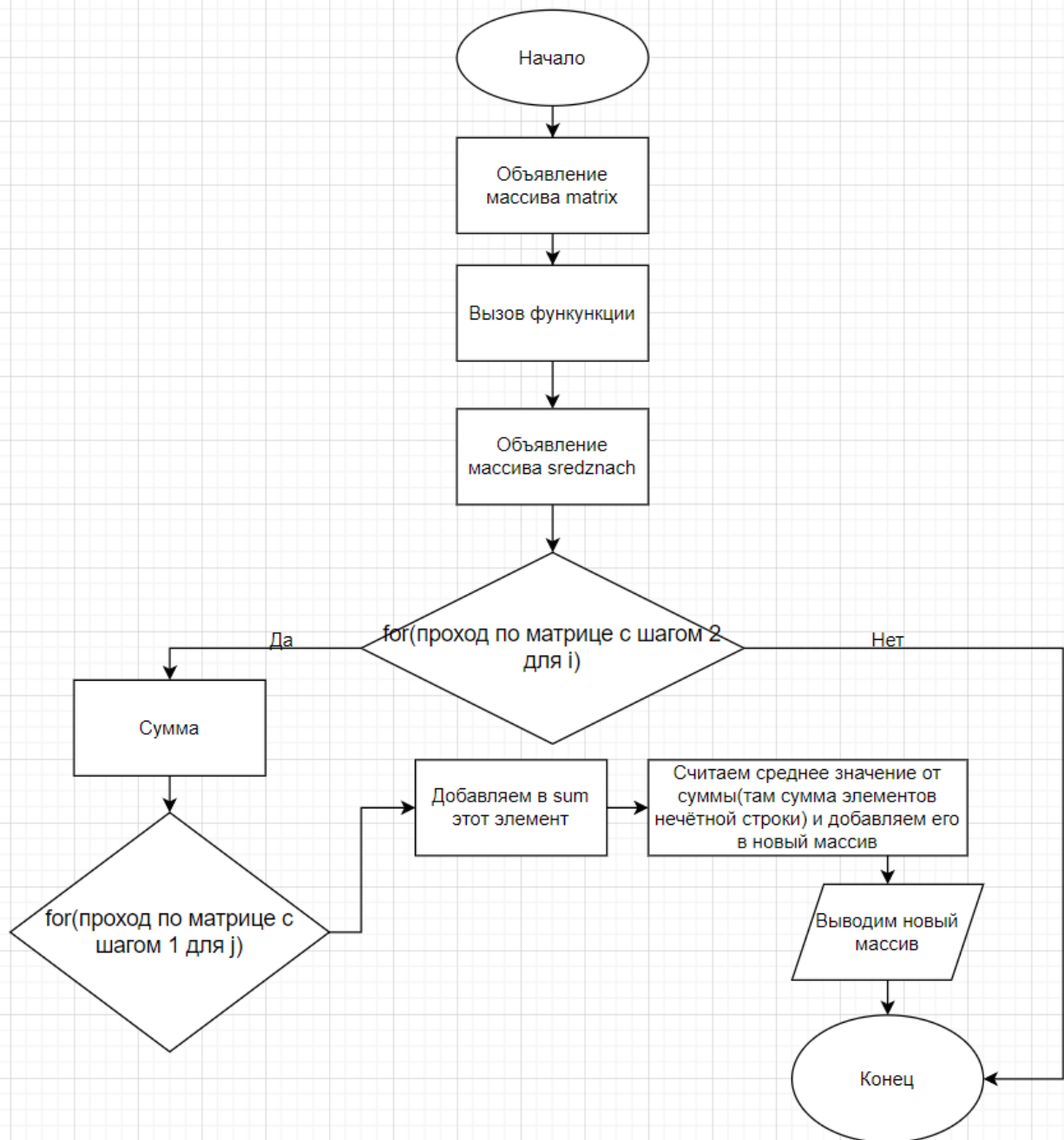


Рисунок 2 – Блок-схема программы.

Тесты:

```
14 ▸ const matrix = [  
15   [1, 2, 3],  
16   [4, 5, 6],  
17   [7, 8, 9],  
18   [10, 11, 12],  
19   [14, 45, 78],  
20   [45, 67, 113]  
21 ];
```

Calc
2,8,45.666666666666664

Рисунок 3 – Тест программы.

```
14 ▸ const matrix = [  
15   [1, 2, 3],  
16   [4, 5, 6],  
17   [7, 8, 9],  
18   [10, 11, 12],  
19   [14, 45, 78],  
20   [45, 67, 113],  
21   [1122, 34, 5678]  
22 ];
```

Calc
2,8,45.666666666666664,2278

Рисунок 4 – Тест программы.

```
14 ▸ const matrix = [  
15   [1, 2, 3],  
16   [4, 5, 6],  
17   [7, 8, 9],  
18   [10, 11, 12],  
19 ];  
20
```

Calc
2,8

Рисунок 5 – Тест программы.

Задание №2. Дана квадратная матрица A порядка M . Найти максимальный элемент для каждой ее диагонали, параллельной побочной (начиная с одноэлементной диагонали $A_{1,1}$).

Текст программы:

HTML ▼

```
1 <button id="btnCalc" onclick="calculate()">
2   Calc
3 </button>
4
5 <div id="divResult"> </div>
```

JavaScript + Без библиотеки (чистый JS) ▼

```
1 function calculate(matrix) {
2
3   const maxEl = [];
4
5   for (let i = 0; i < matrix.length; i++) {
6     let max = matrix[i][0];
7     for (let j = 0; j < matrix.length - i; j++) {
8       if (matrix[j][j + i] > max) {
9         max = matrix[j][j + i];
10      }
11    }
12  }
13
14   maxEl.push(max);
15 }
16
17 return maxEl;
18 }
19 const matrix = [
20   [1, 2, 3],
21   [4, 5, 6],
22   [7, 8, 9]
23 ];
24
25
26 const rez = calculate(matrix);
27 document.getElementById("divResult").innerHTML = rez;
```

Рисунок 6 – Код программы.

Блок-схема:

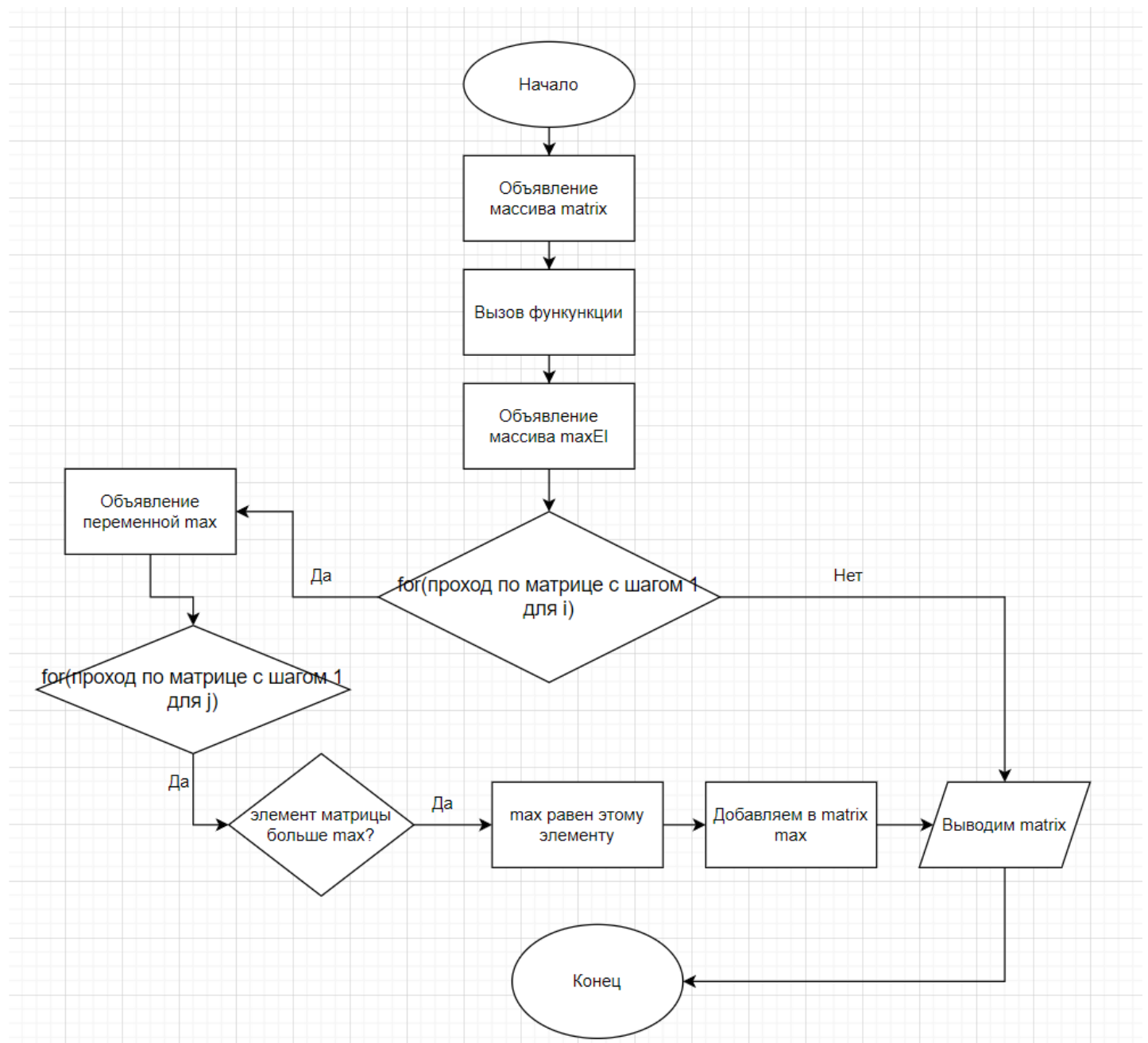


Рисунок 7 – Блок-схема программы.

Тесты:

```
19 ▾ const matrix = [  
20     [1, 2, 3],  
21     [4, 5, 6],  
22     [7, 8, 9]  
23 ];
```

Calc
9,6,7

Рисунок 8 – Тест программы.

```
19 ▾ const matrix = [  
20     [1, 2, 3, 67],  
21     [4, 5, 6, 1267],  
22     [7, 8, 9, 1867],  
23     [10, 11, 12, 13]  
24 ];
```

Calc
13,1867,1267,67

Рисунок 9 – Тест программы.

```
19 ▾ const matrix = [  
20     [1,34, 2, 3, 67],  
21     [4, 5, 56, 6, 1267],  
22     [7, 8, 9, 89, 1867],  
23     [10, 11, 12, 13, 789],  
24     [14, 15, 16, 17777,175]  
25 ];
```

Calc
175,789,1867,1267,67

Рисунок 10 – Тест программы.

```
19 ▾ const matrix = [  
20     [1,34, 2, 3, 67],  
21     [4, 5, 56, 6, 1267],  
22     [7, 8, 9, 89, 1867],  
23     [10, 11, 12, 13, 789],  
24     [1456565, 15, 16, 17777,175]  
25 ];
```

Calc
175,789,1867,1267,1456565

Рисунок 11 – Тест программы.

Вывод

В ходе выполнения лабораторной работы №10 были рассмотрены и решены две интересные задачи, связанные с обработкой матриц.

В первом задании требовалось найти среднее арифметическое элементов каждой строки матрицы с нечетным номером, не используя условные операторы. Для решения этой задачи был использован подход с использованием циклов и арифметических операций. Благодаря этому удалось эффективно вычислить средние значения для каждой нечетной строки матрицы без прямого использования условных операторов.

Во втором задании требовалось найти максимальный элемент для каждой диагонали матрицы, параллельной побочной. Для решения этой задачи также был применен циклический подход с использованием двух вложенных циклов. Это позволило находить максимальные элементы для каждой диагонали матрицы, начиная с одноэлементной диагонали $A_{1,1}$.