



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК5 «Системы обработки информации»

ПРАКТИЧЕСКАЯ РАБОТА №4

Вариант №1

ТЕМА: Работа с массивами

Выполнил: студент гр.

_____ ()
(Подпись)

(Ф.И.О.)

Проверил:

_____ (Трешневская В. О.)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

Калуга , 202 г.

Цель: формирование практических навыков разработки эффективных алгоритмов обработки массивов

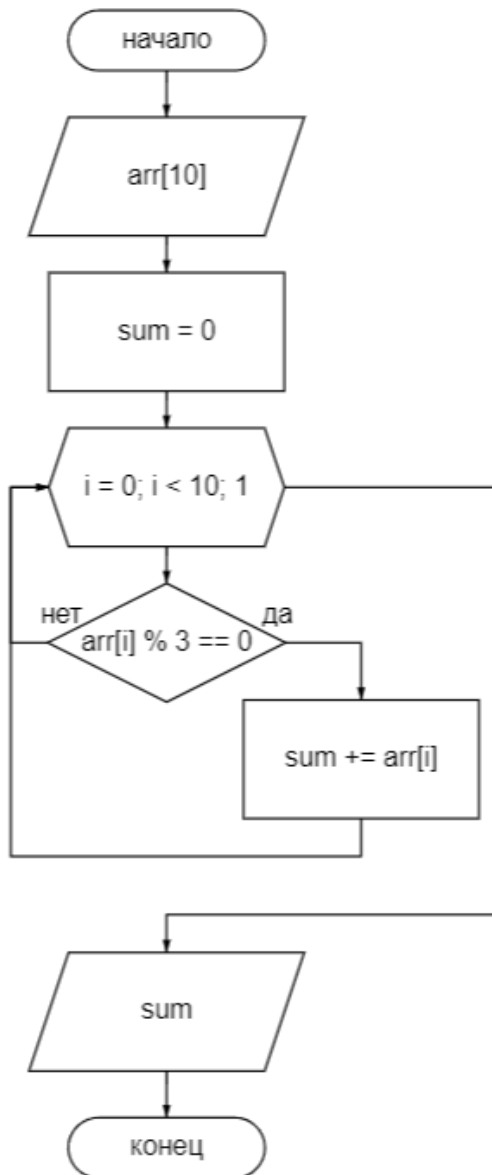
Задачи:

1. Овладеть навыками построения эффективных алгоритмов обработки массивов.
2. Овладеть навыками выбора оптимальных алгоритмов программирования.
3. Овладеть навыками описания основных этапов построения алгоритмов

Порядок выполнения: Вариант 1

Задание 1:

Массив из 10 элементов заполняется с клавиатуры пользователем. Вывести элементы массива на экран. Найти сумму элементов массива кратных 3 и вывести ее на экран.



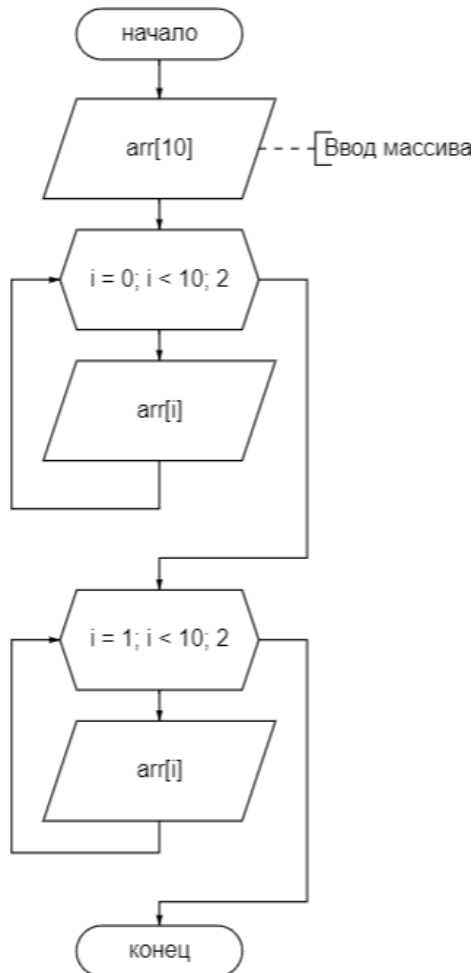
```
1  #include <iostream>
2
3  int main(void)
4  {
5      int arr[10];
6
7      for (int i = 0; i < 10; i++)
8          std::cin >> arr[i];
9
10     int sum = 0;
11     for (int i = 0; i < 10; i++)
12         if (arr[i] % 3 == 0)
13             sum += arr[i];
14
15     std::cout << "Sum: " << sum << std::endl;
16
17     return 0;
18 }
```

1 2 3 4 5 6 7 8 9 0
Sum: 18

Задание 2:

Массив заполняется пользователем с клавиатуры. Вывести на экран сначала все элементы с четными индексами, затем с нечетными.

Пример: Массив: 1 4 5 -6 2 0 7 -9 Результат: 4 -6 0 -9 1 5 2 7.

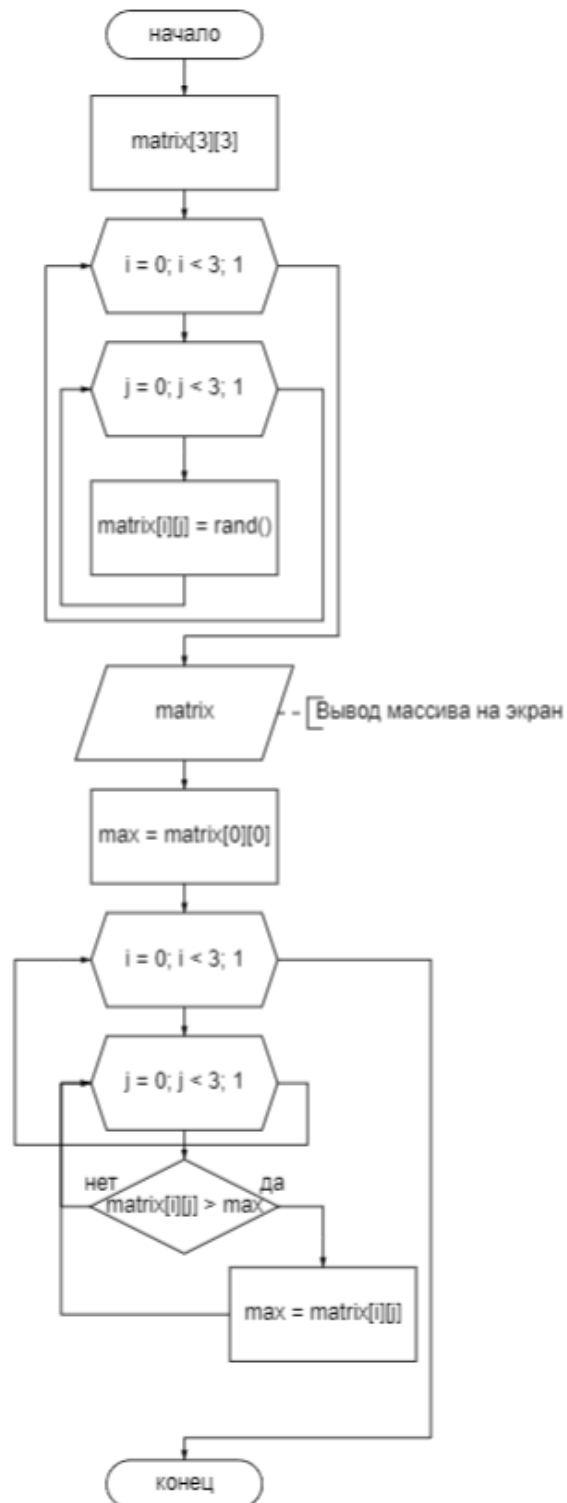


```
1  #include <iostream>
2
3  ✓ int main(void)
4  {
5      int arr[10];
6
7      for (int i = 0; i < 10; i++)
8          std::cin >> arr[i];
9
10     std::cout << "Even index: ";
11     for (int i = 0; i < 10; i += 2)
12         std::cout << arr[i] << ' ';
13
14     std::cout << std::endl;
15     std::cout << "Odd index: ";
16     for (int i = 1; i < 10; i += 2)
17         std::cout << arr[i] << ' ';
18
19     return 0;
20 }
```

```
1 2 3 4 5 6 7 8 9 0
Even index: 1 3 5 7 9
Odd index: 2 4 6 8 0
```

Задание 3:

Матрица 3x3 заполняется случайными числами. Найти максимальный элемент матрицы и вывести на экран.

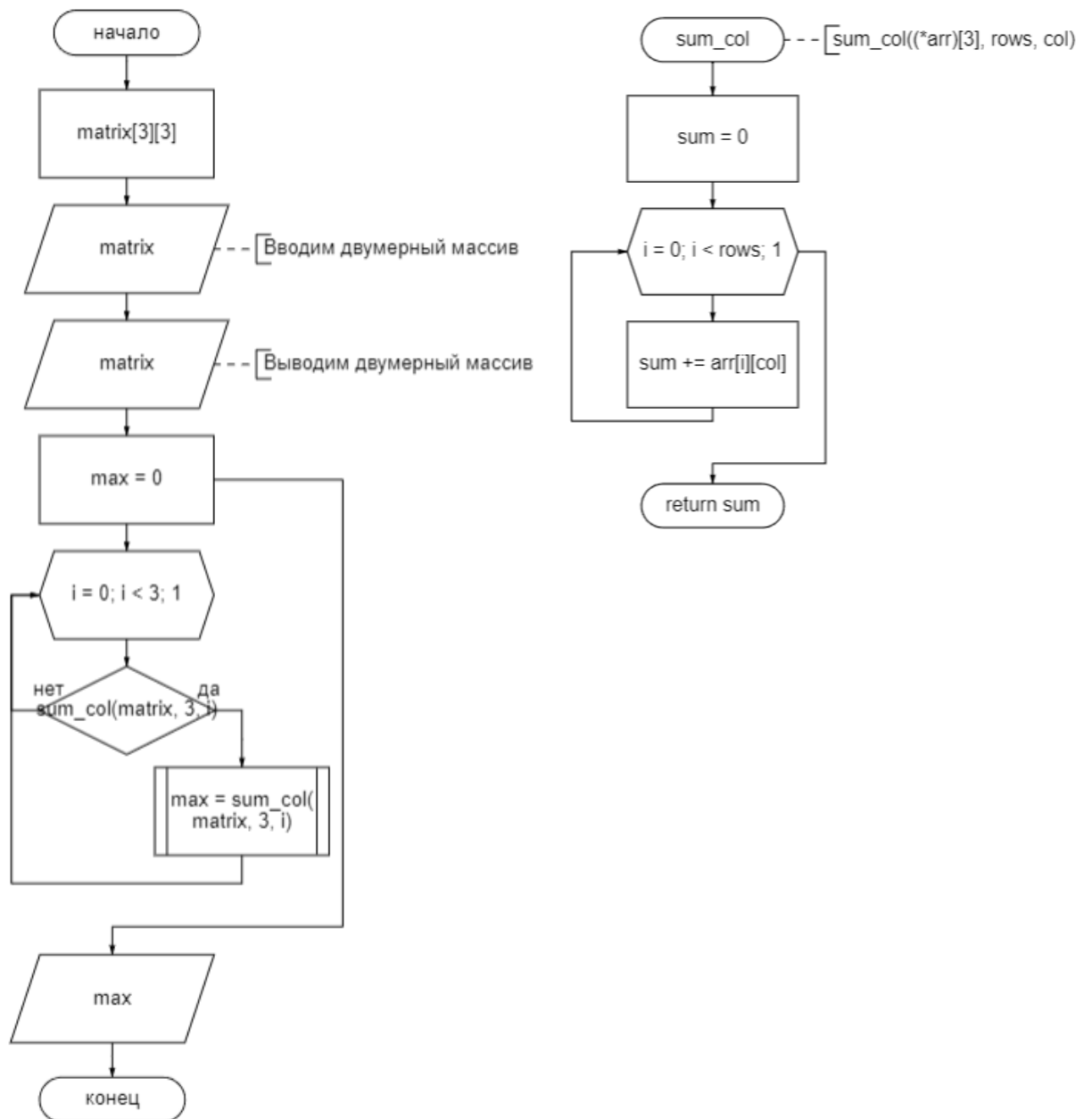


```
1 #include <iostream>
2 #include <iomanip>
3 #include <random>
4
5 int main(void)
6 {
7     int matrix[3][3];
8
9     for (int i = 0; i < 3; i++)
10         for (int j = 0; j < 3; j++)
11             matrix[i][j] = rand();
12
13     for (int i = 0; i < 3; i++)
14     {
15         for (int j = 0; j < 3; j++)
16             std::cout << std::setw(6) << matrix[i][j] << ' ';
17         std::cout << std::endl;
18     }
19
20     int max = matrix[0][0];
21     for (int i = 0; i < 3; i++)
22         for (int j = 0; j < 3; j++)
23             if (matrix[i][j] > max)
24                 max = matrix[i][j];
25
26     std::cout << "Max value: " << max << std::endl;
27
28     return 0;
29 }
```

```
41  18467  6334
26500  19169  15724
11478  29358  26962
Max value: 29358
```

Задание 4:

Задана матрица неотрицательных чисел. Посчитать сумму элементов в каждом столбце. Определить, какой столбец содержит максимальную сумму.



```

1  #include <iostream>
2  #include <iomanip>
3
4  int sum_col(const int (*arr)[3], int rows, int col)
5  {
6      int sum = 0;
7      for (int i = 0; i < rows; i++)
8          sum += arr[i][col];
9
10     return sum;
11 }
12
13 int main(void)
14 {
15     int matrix[3][3];
16
17     for (int i = 0; i < 3; i++)
18         for (int j = 0; j < 3; j++)
19             std::cin >> matrix[i][j];
20
21     for (int i = 0; i < 3; i++)
22     {
23         for (int j = 0; j < 3; j++)
24             std::cout << matrix[i][j] << ' ';
25         std::cout << std::endl;
26     }
27     std::cout << "-----" << std::endl;
28
29     int max = 0;
30     for (int i = 0; i < 3; i++)
31     {
32         if (sum_col(matrix, 3, i) > max)
33             max = sum_col(matrix, 3, i);
34
35         std::cout << sum_col(matrix, 3, i) << ' ';
36     }

```

1 2 3

4 5 6

7 8 9

12 15 18

Max value of sum_col: 18

Вывод: были сформированы практические навыки по разработки эффективных алгоритмов обработки массивов.