

Частное образовательное учреждение высшего образования  
“ЮЖНЫЙ УНИВЕРСИТЕТ (ИУБиП)”

### **Индивидуальное задание №1**

по дисциплине: «Практикум по программированию»

Тема 2. Функции  
Вариант 15.

Работу выполнил:  
студент группы ИД 401  
Недвигин С. В.

Проверил:  
преподаватель к.т.н. Сафонова С. А.

Задание: найти суммы всех положительных элементов массива, имеющих четные индексы. Выполнить для двух массивов.

Задание 2: сформировать массив из минимальных элементов столбцов матрицы. Выполнить для матриц А (6,6) и В (4,4).

Листинг кода.

```
#include <iostream>

void inputArray(int arr[], int size) {
    std::cout << "input array:\n";
    for (int i = 0; i < size; ++i) {
        std::cout << "arr[" << i << "]: ";
        std::cin >> arr[i];
    }
}

void printArray(const int arr[], int size) {
    std::cout << "array:\n";
    for (int i = 0; i < size; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}

int sumPositiveEvenIndices(const int arr[], int size) {
    int sum = 0;
    for (int i = 0; i < size; i += 2) {
        if (arr[i] > 0) {
            sum += arr[i];
        }
    }
    return sum;
}

int main() {
    const int arraySize = 6;
    int array1[arraySize];
    int array2[arraySize];

    inputArray(array1, arraySize);
    inputArray(array2, arraySize);
}
```

```

    printArray(array1, arraySize);
    printArray(array2, arraySize);

    int sum1 = sumPositiveEvenIndices(array1, arraySize);
    int sum2 = sumPositiveEvenIndices(array2, arraySize);

    std::cout << "sum1: " << sum1 << std::endl;
    std::cout << "sum2: " << sum2 << std::endl;

    return 0;
}

```

---

```

#include <iostream>
#include <iomanip> // Для управления выводом чисел с плавающей точкой
#include <limits> // Для ограничения ввода

// Шаблон функции для ввода матрицы
template <typename T>
void inputMatrix(T matrix[][6], int rows, int cols) {
    std::cout << "Введите элементы матрицы " << rows << "x" << cols << ":\n";
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            std::cout << "matrix[" << i << "][" << j << "]: ";
            std::cin >> matrix[i][j];
        }
    }
}

// Шаблон функции для вывода матрицы
template <typename T>
void printMatrix(const T matrix[][6], int rows, int cols) {
    std::cout << "Матрица " << rows << "x" << cols << ":\n";
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            std::cout << std::setw(5) << matrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
}

```

```
// Шаблон функции для формирования массива из минимальных элементов столбцов
```

```
template <typename T>
```

```
void minElementsArray(const T matrix[][6], int rows, int cols, T resultArray[]) {  
    for (int j = 0; j < cols; ++j) {  
        T minElement = matrix[0][j];  
        for (int i = 1; i < rows; ++i) {  
            if (matrix[i][j] < minElement) {  
                minElement = matrix[i][j];  
            }  
        }  
        resultArray[j] = minElement;  
    }  
}
```

```
// Функция для вывода массива целых чисел
```

```
void printIntArray(const int arr[], int size) {  
    std::cout << "Массив из минимальных элементов столбцов:\n";  
    for (int i = 0; i < size; ++i) {  
        std::cout << arr[i] << " ";  
    }  
    std::cout << std::endl;  
}
```

```
// Функция для вывода массива вещественных чисел
```

```
void printDoubleArray(const double arr[], int size) {  
    std::cout << "Массив из минимальных элементов столбцов:\n";  
    for (int i = 0; i < size; ++i) {  
        std::cout << std::fixed << std::setprecision(2) << arr[i] << " ";  
    }  
    std::cout << std::endl;  
}
```

```
int main() {
```

```
    const int rowsA = 6, colsA = 6;
```

```
    const int rowsB = 4, colsB = 4;
```

```
    // Матрицы
```

```
int matrixA[rowsA][6];
double matrixB[rowsB][6];

// Ввод матриц
inputMatrix(matrixA, rowsA, colsA);
inputMatrix(matrixB, rowsB, colsB);

// Вывод матриц
printMatrix(matrixA, rowsA, colsA);
printMatrix(matrixB, rowsB, colsB);

// Массивы минимальных элементов столбцов
int resultArrayA[colsA];
double resultArrayB[colsB];

// Вызов функции для формирования массива из минимальных элементов
столбцов
minElementsArray(matrixA, rowsA, colsA, resultArrayA);
minElementsArray(matrixB, rowsB, colsB, resultArrayB);

// Вывод массивов
printIntArray(resultArrayA, colsA);
printDoubleArray(resultArrayB, colsB);

return 0;
}
```

Скриншот выполнения работы представлен на рисунках 1 и 2.

```
1 #include <iostream>
2
3 void inputArray(int arr[], int size) {
4     std::cout << "input array:\n";
5     for (int i = 0; i < size; ++i) {
6         std::cout << "arr[" << i << "] ";
7         std::cin >> arr[i];
8     }
9 }
10
11 void printArray(const int arr[], int size) {
12     std::cout << "array:\n";
13     for (int i = 0; i < size; ++i) {
14         std::cout << arr[i] << " ";
15     }
16     std::cout << std::endl;
17 }
18
19 int sumPositiveEvenIndices(const int arr[], int size) {
20     int sum = 0;
21     for (int i = 0; i < size; i += 2) {
22         if (arr[i] > 0) {
23             sum += arr[i];
24         }
25     }
26     return sum;
27 }
28
29 int main() {
30     const int arraySize = 6;
31     int array1[arraySize];
32     int array2[arraySize];
33
34     inputArray(array1, arraySize);
35     inputArray(array2, arraySize);
36
37     printArray(array1, arraySize);
38     printArray(array2, arraySize);
39
40     int sum1 = sumPositiveEvenIndices(array1, arraySize);
41     int sum2 = sumPositiveEvenIndices(array2, arraySize);
42
43     std::cout << "sum1: " << sum1 << std::endl;
44     std::cout << "sum2: " << sum2 << std::endl;
45
46     return 0;
47 }
```

Console output:

```
input array:
arr[0]: 2
arr[1]: 4
arr[2]: 6
arr[3]: 8
arr[4]: 9
arr[5]: 2
input array:
arr[0]: 4
arr[1]: 5
arr[2]: 3
arr[3]: 1
arr[4]: 7
arr[5]: 8
array:
2 4 6 8 9 2
array:
4 5 3 1 7 8
sum1: 9
sum2: 14
```

Рисунок 1

```
1 #include <iostream>
2 #include <conio.h> // для функции getch() и _getch()
3 #include <limits> // для INT_MIN
4
5 // Функция для ввода матрицы
6 template <typename T>
7 void inputMatrix(T matrix[][6], int rows, int cols) {
8     std::cout << "Введите элементы матрицы " << rows << "x" << cols << ":\n";
9     for (int i = 0; i < rows; ++i) {
10         for (int j = 0; j < cols; ++j) {
11             std::cout << "matrix[" << i << "][" << j << "] ";
12             std::cin >> matrix[i][j];
13         }
14     }
15 }
16
17 // Функция для вывода матрицы
18 template <typename T>
19 void printMatrix(const T matrix[][6], int rows, int cols) {
20     std::cout << "Матрица " << rows << "x" << cols << ":\n";
21     for (int i = 0; i < rows; ++i) {
22         for (int j = 0; j < cols; ++j) {
23             std::cout << std::setw(5) << matrix[i][j] << " ";
24         }
25         std::cout << std::endl;
26     }
27 }
28
29 // Функция для формирования массива из минимальных элементов столбцов
30 template <typename T>
31 void minElementArray(const T matrix[][6], int rows, int cols, T resultArray[]) {
32     for (int j = 0; j < cols; ++j) {
33         T minElement = matrix[0][j];
34         for (int i = 1; i < rows; ++i) {
35             if (matrix[i][j] < minElement) {
36                 minElement = matrix[i][j];
37             }
38         }
39         resultArray[j] = minElement;
40     }
41 }
42
43 // Функция для вывода массива минимальных элементов
44 void printIntArray(const int arr[], int size) {
45     std::cout << "Массив из минимальных элементов столбцов:\n";
46     for (int i = 0; i < size; ++i) {
47         std::cout << arr[i] << " ";
48     }
49     std::cout << std::endl;
50 }
51
52 // Функция для вывода массива минимальных элементов столбцов
53 void printDoubleArray(const double arr[], int size) {
54     std::cout << "Массив из минимальных элементов столбцов:\n";
55     for (int i = 0; i < size; ++i) {
56         std::cout << std::fixed << std::setprecision(2) << arr[i] << " ";
57     }
58 }
```

Console output:

```
Введите элементы матрицы 5x6:
matrix[0][0]: 1
matrix[0][1]: 2
matrix[0][2]: 3
matrix[0][3]: 4
matrix[0][4]: 5
matrix[0][5]: 6
matrix[1][0]: 7
matrix[1][1]: 9
matrix[1][2]: 1
matrix[1][3]: 2
matrix[1][4]: 3
matrix[1][5]: 4
matrix[2][0]: 6
matrix[2][1]: 5
matrix[2][2]: 6
matrix[2][3]: 7
matrix[2][4]: 8
matrix[2][5]: 9
matrix[3][0]: 1
matrix[3][1]: 2
matrix[3][2]: 3
matrix[3][3]: 4
matrix[3][4]: 5
matrix[3][5]: 6
matrix[4][0]: 7
matrix[4][1]: 8
matrix[4][2]: 9
matrix[4][3]: 1
matrix[4][4]: 2
matrix[4][5]: 3
matrix[5][0]: 4
matrix[5][1]: 5
matrix[5][2]: 6
matrix[5][3]: 7
matrix[5][4]: 8
matrix[5][5]: 9
Матрица 5x6:
1 2 3 4 5 6
7 8 9 1 2 3
4 5 6 7 8 9
1 2 3 4 5 6
7 8 9 1 2 3
4 5 6 7 8 9
Матрица 5x6:
1 2 3 4
5 6 7 8
9 1 2 3
4 5 6 7
Массив из минимальных элементов столбцов:
1 2 3 1 2 3
Массив из минимальных элементов столбцов:
1.00 1.00 2.00 3.00
```

Рисунок 2

Вывод: в ходе выполнения работы освоены программы простейшей структуры; приобретены навыки написания функций с использованием шаблонов и вызова функций через указатели.