



МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Санкт-Петербургский государственный технологический институт  
(технический университет)»

УГСН	09.00.00	Информатика и вычислительная техника
Уровень образования		Высшее образование – бакалавриат
Форма обучения		Очная
Факультет		Информационных технологий и управления
Кафедра		Систем автоматизированного проектирования и управления
Учебная дисциплина		Информационные технологии и программирование
Курс	I	Группа 4304

**Отчёт по контрольной работе № 3**

**Вариант № 26**

Исполнитель:  
обучающийся  
группы 4304

\_\_\_\_\_  
(дата, подпись)

Рыбник Всеволод Сергеевич

Проверил:

\_\_\_\_\_  
(дата, подпись)

Корниенко Иван Григорьевич

Макарук Роман Валерьевич

Федин Алексей Константинович

Санкт-Петербург  
2023

## СОДЕРЖАНИЕ

1 Задание №1.....	3
1.1 Цель работы.....	3
1.2 Постановка задачи.....	3
1.3 Описание хода выполнения.....	3
1.4 Блок-схема алгоритма решения задачи.....	3
1.5 Исходный код полученного программного решения.....	5
1.6 Тестирование.....	6
1.7 Выводы по заданию №1.....	6
2 Задание №2.....	7
2.1 Цель работы.....	7
2.2 Постановка задачи.....	7
2.3 Описание хода выполнения.....	7
2.4 Блок-схема алгоритма решения задачи.....	7
2.5 Исходный код полученного программного решения.....	9
2.6 Тестирование.....	13
2.7 Выводы по заданию №2.....	13
3 Задание №3.....	15
3.1 Цель работы.....	15
3.2 Постановка задачи.....	15
3.3 Описание хода выполнения.....	15
3.4 Блок-схема алгоритма решения задачи.....	15
3.5 Исходный код полученного программного решения.....	17
3.6 Тестирование.....	20
3.7 Выводы по заданию №3.....	21

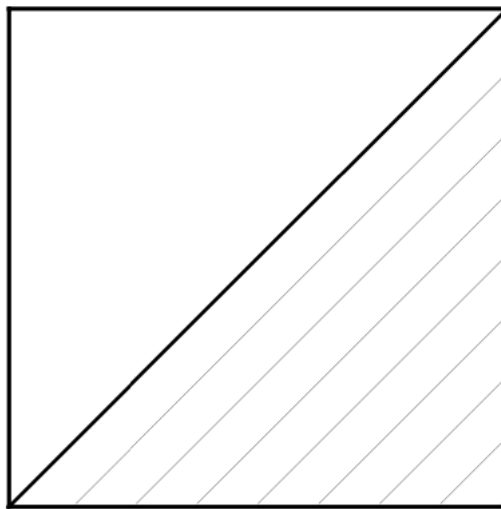
## 1 Задание №1

### 1.1 Цель работы

Динамическое распределение памяти.

### 1.2 Постановка задачи

Разработать программу решения поставленной задачи: «Даны числа  $n$  и  $m$ , действительная матрица  $A_{n \times m}$ . Найти и вывести наименьшее  $E_{mn}$  и наибольшее  $E_{mx}$  из значений элементов, расположенных в заштрихованной части матрицы:



Вывести полученную матрицу  $A$ .»

### 1.3 Описание хода выполнения

В ходе выполнения задачи была использована структура, выполняющая функции двумерного массива — матрицы.

### 1.4 Блок-схема алгоритма решения задачи

Блок-схема алгоритма №3.1:

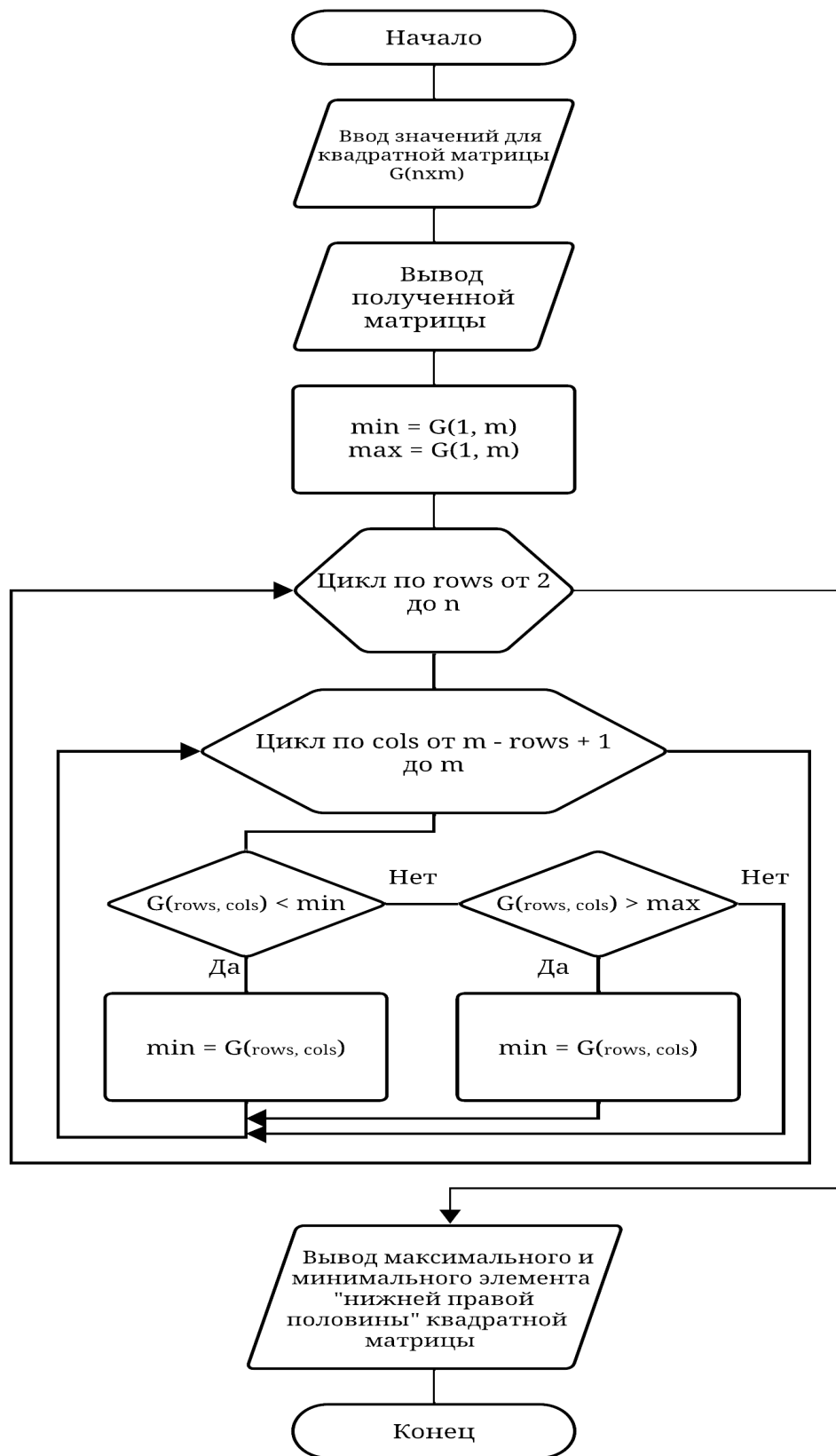


Рисунок 1 – Блок-схема алгоритма решения задачи №1

### **1.5 Исходный код полученного программного решения**

### **1.6 Тестирование**

Результат тестирования приведён на рисунке 2.

Рисунок 2 – Экранная копия результата работы разработанной программы задания №1

### **1.7 Выводы по заданию №1**

## **2 Задание №2**

### **2.1 Цель работы**

Работа с битами.

### **2.2 Постановка задачи**

Разработать решение поставленной задачи: «Дана последовательность из 8 символов. Сравнить их младший и старший биты. Если они равны, то заменить старший нулём, младший – единицей, иначе заменить старший бит единицей, младший – нулём.»

### **2.3 Описание хода выполнения**

### **2.4 Блок-схема алгоритма решения задачи**

На рисунке 3 представлена блок-схема алгоритма решения задачи №2.

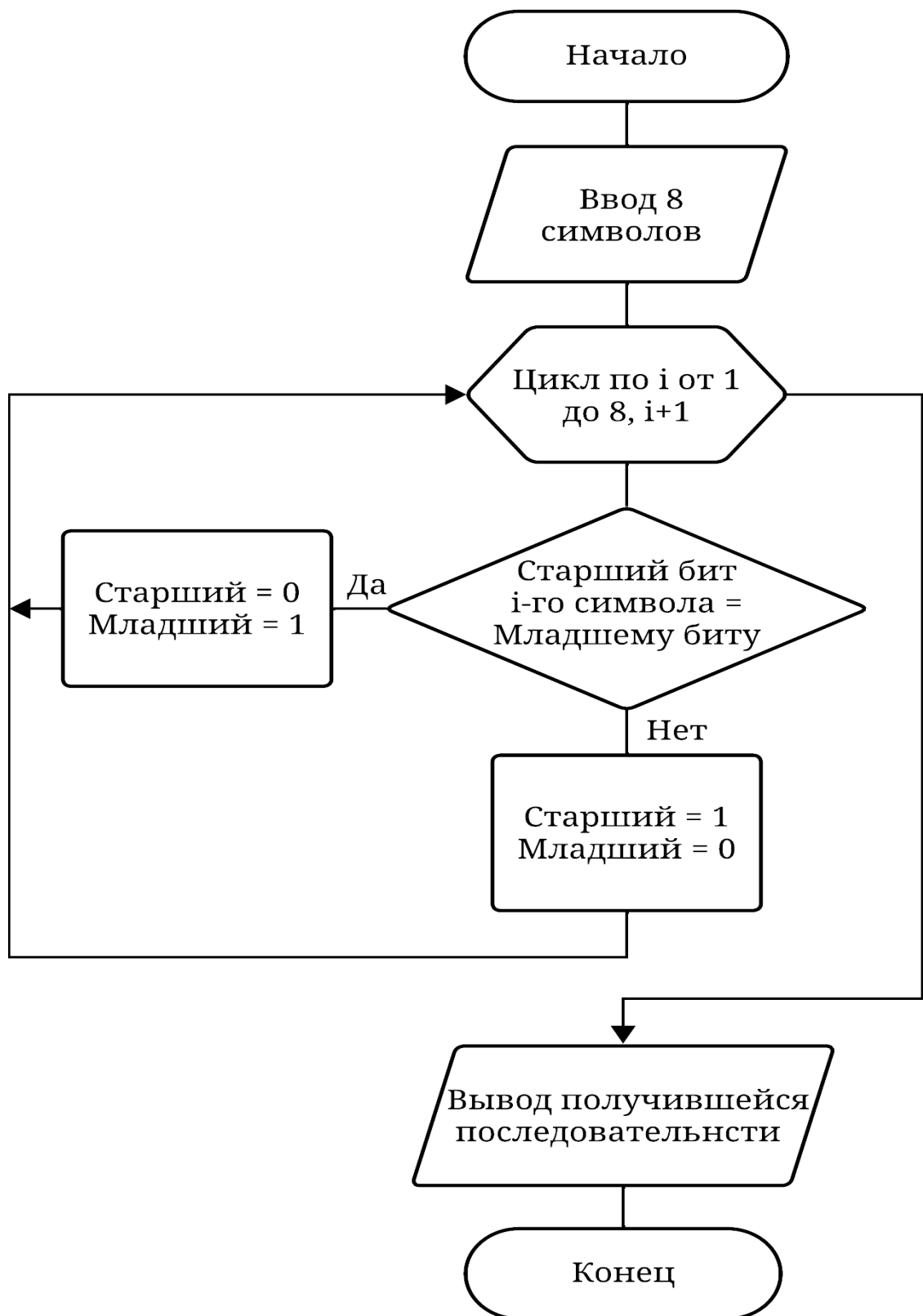


Рисунок 3 – Блок-схема алгоритма решения задачи №2

## **2.5 Исходный код полученного программного решения**

Код основного файла – `second_task.c`:

## **2.6 Тестирование**

Тестирование было проведено на рандомных значениях:

Рисунок 4 – Экранная копия результата работы разработанной программы задания 2

## **2.7 Выводы по заданию №2**



### **3 Задание №3**

#### **3.1 Цель работы**

Разобраться в функционале указателей на функции.

#### **3.2 Постановка задачи**

Разработать решение поставленной задачи: «Функция `fold` суммирует все числа в массиве целочисленных чисел с использованием функции `sum`, получает в качестве аргументов указатель на исходный массив, размер массива, указатель на функцию `sum` от двух аргументов и возвращает сумму всех элементов массива. Функция `sum` производит сложение двух аргументов и возвращает их сумму.»

#### **3.3 Описание хода выполнения**

Для реализации поставленной задачи были изучены указатели на функции в Objective C.

#### **3.4 Блок-схема алгоритма решения задачи**

Блок схема к задаче №3:

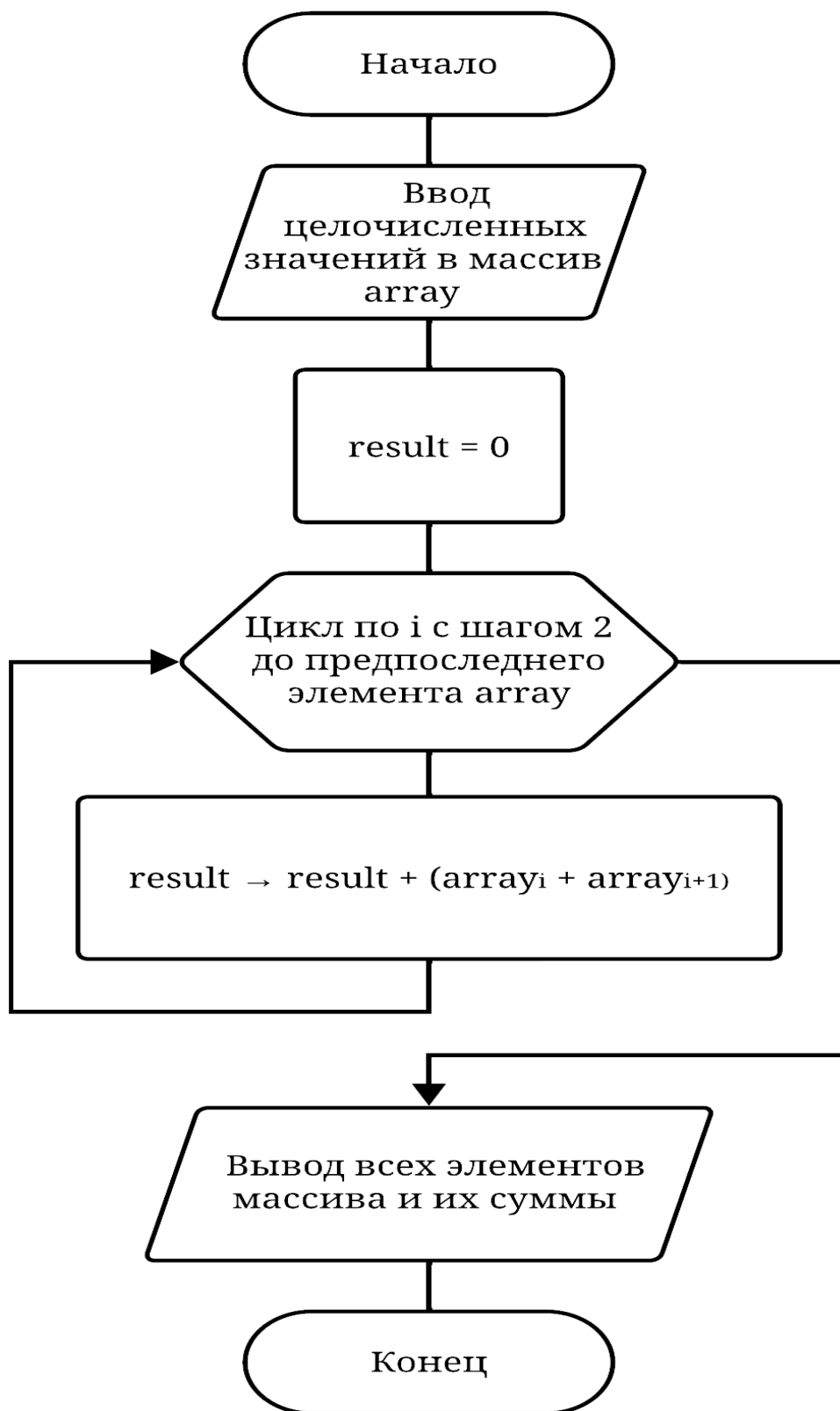


Рисунок 5 – Блок-схема алгоритма решения задачи №3

### **3.5 Исходный код полученного программного решения**

Код основного файла `third_task.c`:

### **3.6 Тестирование**

Тестирование было проведено на рандомных значениях:

Рисунок 6 – Экранная копия результата работы разработанной программы задания №3

### **3.7 Выводы по заданию №3**