

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт сервиса, туризма и дизайна (филиал) СКФУ в г. Пятигорске  
Кафедра информационной безопасности, систем и технологий

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
по выполнению курсовых работ  
по дисциплине  
**ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

Направление подготовки	10.03.01 Информационная безопасность
Профиль	Комплексная защита объектов информатизации
Квалификация выпускника	бакалавр
Форма обучения	очная
Учебный план	2020 г.

Пятигорск 2020 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. СТРУКТУРА КУРСОВОЙ РАБОТЫ.....	4
2. ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.....	5
3. ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ.....	6
4. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ.....	14
4.1 Пример выполнения задания 1.....	14
4.2 Пример выполнения задания 2.....	18
ЗАКЛЮЧЕНИЕ.....	22

## **ВВЕДЕНИЕ**

Целью курсовой работы по дисциплине «Основы алгоритмизации и программирования» является закрепление навыков построения алгоритмов, написания и отладки программ на языке C# в среде Visual Studio 2012 – 2017.

Приобретенные навыки будут необходимы в дальнейшем для выполнения лабораторных работ по таким дисциплинам, как «Математические основы криптологии», «Криптографические методы защиты информации» и другим.

Курсовая работа состоит из двух заданий: первое – на обработку числовых массивов, второе – на обработку символьной информации и работу с текстовыми файлами.

Для каждого задания студент должен привести описание алгоритма решения задачи, таблицу используемых идентификаторов, код программы и результаты тестирования программы в виде скриншотов окон вывода.

Во введении к курсовой работе необходимо описать цели и задачи работы, ее краткое содержание, а также привести краткие сведения о типовых структурах алгоритмов и наиболее популярных языках и средах программирования.

Оценка за курсовую работу будет зависеть от правильности выполненных заданий, оформления пояснительной записки и ответов студента на вопросы преподавателя в ходе защиты работы.

Методические рекомендации к курсовой работе содержат варианты заданий, описание структуры курсовой работы, требования к оформлению работы и примеры выполнения заданий.

## 1. СТРУКТУРА КУРСОВОЙ РАБОТЫ

Пояснительная записка к курсовой работе должна содержать следующие разделы:

ТИТУЛЬНЫЙ ЛИСТ;

ЛИСТ ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ;

СОДЕРЖАНИЕ;

ВВЕДЕНИЕ;

1. ЗАДАНИЕ 1;

1.1 Постановка задачи 1;

1.2 Таблица идентификаторов (спецификации) для задачи 1;

1.3 Описание алгоритма решения задачи 1;

1.4 Листинг программы 1;

1.5 Результаты тестирования программы 1;

2. ЗАДАНИЕ 2;

2.1 Постановка задачи 2;

2.2 Таблица идентификаторов (спецификации) для задачи 2;

2.3 Описание алгоритма решения задачи 2;

2.4 Листинг программы 2;

2.5 Результаты тестирования программы 2;

ЗАКЛЮЧЕНИЕ;

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.

Каждый раздел должен начинаться с новой страницы.

Формы *титульного листа* и *листа задания* можно получить на кафедре ИБСиТ.

*Содержание* включает наименование всех разделов курсовой работы, а также подразделов и пунктов, если они имеют наименование, с указанием номеров страниц, на которых размещается начало материала разделов, подразделов и пунктов.

Во *введении* к курсовой работе необходимо описать цели и задачи работы, ее краткое содержание, а так же привести краткие сведения о типовых структурах алгоритмов и наиболее популярных языках и средах программирования. Во *введении* дается краткий анализ возможных методов решения поставленной задачи, например, использование различных систем и языков программирования. Указываются литературные источники, по которым делается обзор, позволяющий судить, насколько полно изучена литература по методике решения поставленной задачи.

Объем введения – 2 - 4 страницы. Ни в *Содержании*, ни в тексте оно не обозначается цифрами, являясь самостоятельной частью работы.

Содержание первого и второго разделов пояснительной записки должно соответствовать приводимым в данных рекомендациях примерам выполнения и оформления заданий.

*Заключение* должно содержать основные результаты выполнения работы.

*Список используемых источников* содержит перечень литературных и интернет источников, использованных при выполнении курсовой работы.

Указываются только те источники, на которые имеются ссылки в тексте пояснительной записки. Список формируется в алфавитном порядке. Количество наименований источников – не менее 5.

## **2. ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

Нормативный объем пояснительной записки к курсовой работе – не менее **15** страниц печатного текста, включая титульный лист.

Пояснительная записка подготавливается на ЭВМ с использованием средств Microsoft Office и распечатывается на листах формата А4 с полями: левое - 30, верхнее –20, нижнее –20, правое - 15 мм. Шрифт: 14 пунктов, Times New Roman, межстрочный промежуток - полуторный.

*Пояснительная записка и диск с программами должны быть упакованы в скоросшиватель.*

Текст пояснительной записки должен быть разбит на разделы. Разделы имеют порядковую нумерацию в пределах всей работы и обозначаются арабскими цифрами с точкой в конце, например: 1.; 2. и т.д. Введение, заключение, список используемых источников и приложения не нумеруются и печатаются с нового листа.

Все страницы пояснительной записки должны быть пронумерованы. Номер страницы ставится внизу страницы по центру. На первых трех листах номер страницы не ставится.

Цифровой материал оформляется в виде таблиц. Таблицы размещают после первого упоминания о них в тексте таким образом, чтобы их можно было читать без поворота работы или с поворотом по часовой стрелке. Каждая таблица должна иметь заголовок. Например: «Таблица 1.1 – Таблица используемых идентификаторов». Нумерация таблиц сквозная в пределах работы или раздела. Если таблица одна, то она не нумеруется, а в тексте слово "таблица" пишется полностью.

Количество иллюстраций в работе должно быть достаточным для пояснения излагаемого текста. Иллюстрации (схемы, графики, рисунки) обозначаются словом "Рисунок" и располагаются так, чтобы их было удобно рассматривать без поворота или с поворотом по часовой стрелке, после первой ссылки на них в тексте. Иллюстрации должны иметь наименование. Иллюстрации нумеруют в пределах работы или раздела арабскими цифрами. Номер и наименование рисунка записываются в строчку под его изображением посередине страницы. Например: «Рисунок 1.1 - Блок-схема алгоритма решения задачи».

### **3. ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ**

Требуется разработать алгоритм и программу для решения двух задач в соответствии с вариантом задания. Для каждой задачи привести таблицу

используемых идентификаторов, словесное описание и блок-схему алгоритма, листинг программы и результаты тестирования программы.

В качестве среды разработки выбрать Microsoft Visual Studio 2012 - 2017, в качестве языка программирования - C#. Вид разрабатываемого приложения – консольное.

Распределение вариантов заданий приведено в таблице 3.1.

Таблица 3.1 – Номера вариантов заданий на курсовую работу

ФИО студента	Номер варианта
Абрамов Станислав	1
Багно Елизавета	2
Бойко Денис	3
Вардазарян Гор	4
Еремин Алексей	5
Калабеков Исмаил	6
Каспарьян Геворк	7
Клюев Дмитрий	8
Кононенко Александр	9
Конопляник Антон	10
Рокшин Владислав	11
Русанов Владислав	12
Тарабанова Мария	13
Тарочкин Артем	14
Трубаев Артём	15
Хадзегова Зарина	16
Янов Тамирлан	17

### 3.1 Варианты задания № 1

#### Вариант 1

Написать программу, которая заполняет матрицу  $A(N,N)$  случайными целыми числами от 0 до  $M$  ( $N$  и  $M$  вводятся с клавиатуры). Затем формирует два одномерных массива  $B$  и  $C$ , причем в массив  $B$  записываются элементы, расположенные на главной диагонали и выше, а в массив  $C$  – элементы матрицы, лежащие ниже главной диагонали. Программа должна выводить все массивы на экран. Осуществить поиск количества вхождений введенного с клавиатуры числа  $X$  в массив  $B$ . Результаты поиска вывести на экран.

#### Вариант 2

Дана квадратная матрица  $A(N,N)$ . Написать программу формирования массива  $B$ , элементами которого являются элементы матрицы  $A$ , расположенные над побочной диагональю. Массивы  $A$  и  $B$  вывести на экран. В массиве  $B$  найти три наименьших элемента и вывести их на экран в порядке возрастания.

#### Вариант 3

Написать программу тестирования знаний студентов по дисциплине Программирование и основы алгоритмизации. Вопросы для тестирования должны выбираться из файла случайным образом. Программа должна осуществлять подсчет правильных ответов и выводить на экран результаты тестирования (общее число вопросов, количество правильных ответов, оценку).

#### Вариант 4

Составить программу, которая заменяет каждый элемент двумерного массива  $A(N,M)$  натуральных чисел на соответствующую запись данного числа в системе счисления с основанием  $K$ ,  $K < 20$ . Числа  $K$ ,  $N$  и  $M$  вводятся с клавиатуры. Исходный и результирующий массивы вывести на экран.

#### Вариант 5

Квадратная матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива. Восстановить исходную матрицу и напечатать по строкам.

#### Вариант 6

Определить, какое число в массиве  $A(N,M)$  встречается чаще всего. Вывести на экран массив, найденное число и количество его повторений.



#### Вариант 7

В одномерном массиве  $A(N)$  найти сумму элементов, находящихся между минимальным и максимальным элементами. Сами минимальный и максимальный элементы в сумму не включать.

#### Вариант 8

Вывести в порядке возрастания цифры, входящие в десятичную запись заданного натурального числа  $N$ .

#### Вариант 9

Из одномерного массива удалить все повторяющиеся элементы (дубликаты) так, чтобы каждое значение встречалось в массиве только один раз.

#### Вариант 10

Найти количество одинаковых элементов в двух заданных одномерных массивах.

#### Вариант 11

Ввести оценки  $N$  учеников по  $K$  предметам ( $N > 10$ ). Определить и вывести на экран количество учеников, не получивших ни одной оценки «5», а также количество неуспевающих учеников. Найти процент учеников, обучающихся на 4 и 5, процент отличников и процент неуспевающих студентов.

#### Вариант 12

Заполните матрицу  $A(N,M)$  случайными целыми числами от 0 до 99. Запишите элементы матрицы  $A$  в матрицу  $B(N,M)$  так, чтобы сначала шли нечетные числа, затем четные. Также вывести количество четных и нечетных чисел в матрице.

#### Вариант 13

Задана матрица  $A(N,N)$  действительных чисел. "Перевернуть" в ней главную и побочную диагонали (переписать цифры в обратном порядке).

#### Вариант 14

Дана целочисленная матрица  $A(N,M)$ . Найти максимум из наименьших элементов столбцов.

### Вариант 15

В матрице  $A(N,M)$  найдите элементы (их позицию), которые являются одновременно минимальными в строке и столбце.

### Вариант 16

В группе учатся  $N$  студентов, каждый из которых получил по четыре оценки за экзамены. Определить количество неуспевающих студентов и средний балл группы.

### Вариант 17

Задан одномерный массив из  $N$  натуральных чисел. Найти и вывести на экран все пары элементов массива, являющиеся взаимно простыми (не имеющие общих делителей, кроме 1).

## 3.2 Варианты задания № 2

### Вариант 1

Дан текст на русском языке (в файле `input.txt`). Определить, сколько раз встречается в нем самое короткое слово. Слово и количество его повторений вывести на экран и в файл `output.txt`.

### Вариант 2

Для заданного в файле `input.txt` текста определить длину содержащейся в нем максимальной последовательности символов, отличных от букв. Найденную последовательность и ее длину вывести на экран и записать в файл `output.txt`.

### Вариант 3

Ученики зашифровывают свои записки, записывая все слова наоборот. Составить программу, зашифровывающую и расшифровывающую сообщение. Текст сообщения содержится в файле `input.txt`. Зашифрованное сообщение записать в файл `output.txt`. Исходное, зашифрованное и расшифрованное сообщения вывести на экран.

### Вариант 4

Написать программу, которая будет исправлять неточности редактирования заданного в файле `input.txt` текста. Программа должна убирать знаки пробела перед знаком препинания, ставить один пробел после знака препинания, исправлять первую букву предложения на заглавную. Отредактированный текст вывести на экран и записать в файл `output.txt`.

### Вариант 5

В файле input.txt содержатся фамилии и имена учеников, общее количество которых не превосходит 100. В первой строке содержится количество учеников, принимавших участие в соревнованиях. Требуется написать программу, которая формирует и печатает уникальный логин для каждого ученика по следующему правилу: если фамилия встречается первый раз, то логин – это данная фамилия, если фамилия встречается второй раз, то логин – это фамилия, в конец которой приписывается число 2 и т.д. Логин учеников записать в файл output.txt. Например, для входной последовательности

5

Иванова Мария

Петров Сергей

Бойцова Екатерина

Петров Иван

Иванова Наташа

будут сформированы следующие логины:

Иванова Петров Бойцова Петров2

Иванова2

### Вариант 6

На городской олимпиаде по информатике участникам было предложено выполнить 3 задания, каждое из которых оценивалось по 25-балльной шкале. Известно, что общее количество участников первого тура олимпиады не превосходит 250 человек. На вход программы подаются сведения о результатах олимпиады: фамилии, имена и баллы. В первой строке вводится количество участников. Например,

2

Петрова Ольга 25 18 16

Калиниченко Иван 14 19 15

Напишите программу, которая будет выводить на экран фамилию и имя участника, набравшего максимальное количество баллов. Если среди остальных участников есть ученики, набравшие такое же количество баллов, то их фамилии и имена также следует вывести. При этом имена и фамилии можно выводить в произвольном порядке.

### Вариант 7

В некотором вузе абитуриенты проходили предварительное тестирование, по результатам которого они могут быть допущены к сдаче вступительных экзаменов в первом потоке. Тестирование проводится по трём предметам, по каждому предмету абитуриент может набрать от 0 до 100 баллов. При этом к сдаче экзаменов в первом потоке допускаются абитуриенты, набравшие по результатам тестирования не менее 30 баллов по каждому из трёх предметов, причём сумма баллов должна быть не менее 140. На вход программы подаются сведения о результатах предварительного тестирования. Известно, что общее количество участников тестирования не превосходит 500. В первой строке вводится количество абитуриентов, принимавших участие в тестировании. Пример входной строки:

2

Романов Владимир 48 39 55

Петрова Ольга 45 28 76

Напишите программу, которая будет выводить на экран фамилии и имена абитуриентов, допущенных к сдаче экзаменов в первом потоке. При этом фамилии должны выводиться в алфавитном порядке.

### Вариант 8

На вход программы подается текст на английском языке, заканчивающийся точкой (в тексте отсутствуют символы, отличные от английских букв и пробелов). Требуется написать программу, которая будет определять и выводить на экран английскую букву, встречающуюся в этом тексте чаще всего, и количество таких букв. Строчные и прописные буквы при этом считаются не различимыми. Если искомым букв несколько, то программа должна выводить на экран первую из них по алфавиту. Например, пусть файл содержит следующую запись:

**It is not a simple task.**

Чаще всего здесь встречаются буквы I, S и T. Следовательно, в данном случае программа должна вывести два символа, разделенных пробелом:

I 3

### Вариант 9

Назовем сложностью предложения сумму количества слов и знаков препинания. Определить сложность заданного в файле input.txt предложения. Вывести на экран количество слов, количество знаков препинания и их сумму.

#### Вариант 10

В файле input.txt задан текст, содержащий не более 1000 символов. Записать в файл output.txt и вывести на экран все слова, содержащие заданную (введенную с клавиатуры) последовательность букв (например, однокоренные слова).

#### Вариант 11

В файле input.txt задан текст, содержащий не более 1000 символов. Записать в файл output.txt и вывести на экран буквы, на которые начинаются слова в тексте, в порядке убывания частоты их употребления.

#### Вариант 12

Список фамилий, разделенных запятыми, задан в файле input.txt в произвольном порядке. Упорядочить его по алфавиту, записать в файл output.txt и вывести на экран.

#### Вариант 13

В файле input.txt задан текст, который содержит слова и целые числа произвольного порядка. Найти и вывести на экран сумму входящих в текст чисел.

#### Вариант 14

Определить, сколько в заданном тексте русских букв, английских букв, цифр и разделителей. Текст задан в файле input.txt. Результат записать в файл output.txt и вывести на экран.

#### Вариант 15

В заданном тексте найти слова, которые являются анаграммами, т.е. читаются одинаково слева направо и справа налево. Текст задан в файле input.txt. Результат записать в файл output.txt и вывести на экран.

#### Вариант 16

В заданном тексте найти слова, начинающиеся и заканчивающиеся на одну и ту же букву. Текст задан в файле input.txt. Результат записать в файл output.txt и вывести на экран.

#### Вариант 17

В заданном тексте найти предложения, содержащие английские слова. Текст задан в файле input.txt. Найденные предложения записать в файл output.txt и вывести на экран.

## 4. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ

### 4.1 Пример выполнения задания 1

#### 4.1.1 Постановка задачи 1

Заполнить массив  $A(N, M)$  случайными целыми числами от 10 до 99.

Переписать в одномерный массив  $B$  элементы массива  $A$ , являющиеся простыми числами, и упорядочить их по возрастанию. Вывести на экран оба массива.

#### 1.1.2- Таблица идентификаторов для задачи 1

Имя идентификатора	Описание идентификатора	Тип идентификатора
N	Число строк исходного массива	целый
M	Число столбцов исходного массива	целый
A	Исходный массив	целый
B	Результирующий массив	целый
prost	Метод, возвращающий значение истина, если аргумент метода является простым числом	логический
flag	Логическая переменная, принимающая значение true, если аргумент k метода prost является простым числом	логический
count	Количество простых чисел в массиве A	целый
i, j	Счетчики цикла	целый
temp	Вспомогательная переменная	целый

#### 1.1.3 Описание алгоритма решения задачи 1

Алгоритм решения задачи можно разбить на следующие этапы:

– создание подпрограммы prost, возвращающей значение «истина», если параметр подпрограммы k является простым числом. Для этого нужно для всех чисел  $i$  от 2 до  $k/2$  проверить, делится ли нацело число  $k$  на  $i$ . Если хотя бы для

одного из этих чисел  $i$  остаток от деления  $k$  на  $i$  равен нулю, число  $k$  не является простым, и подпрограмма возвращает значение «ложь»;

– запись в одномерный массив  $B$  элементов массива  $A$ , являющихся простыми числами;

– сортировка массива  $B$  по возрастанию элементов;

– вывод отсортированного массива на экран.

Блок-схема алгоритма решения задачи приведена на рисунке 4.1 а) – б).

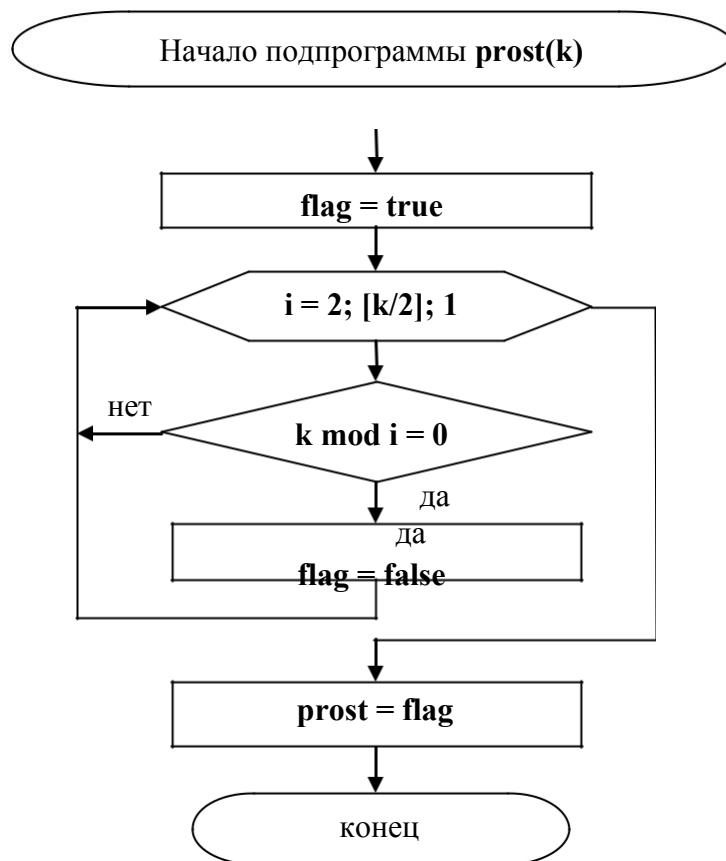


Рисунок 4.1.а – Блок-схема решения задачи 1

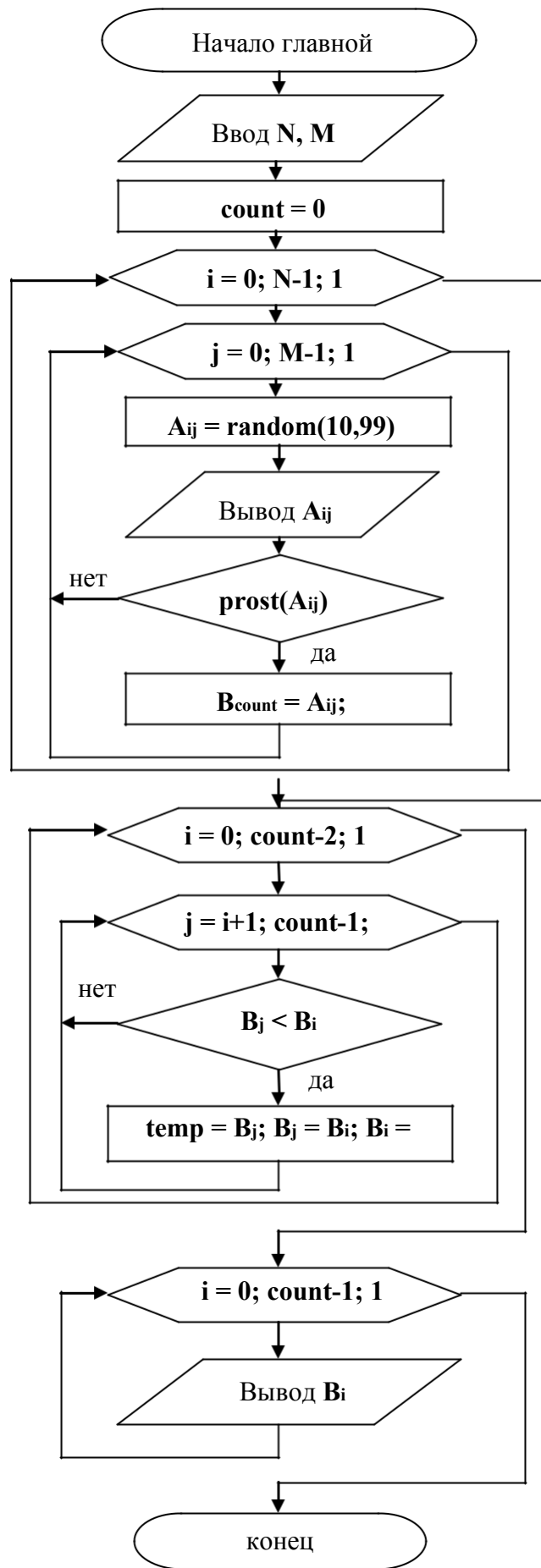


Рисунок 4.1 б – Продолжение блок-схемы решения задачи 1



## 1.1.4 Листинг программы 1

```
using System;
|
|-----|
namespace Числовые_массивы
{
    class Program
    {
        public static Boolean prost(int k) // метод для определения простоты числа k
        {
            Boolean flag = true;
            for (int i = 2; i <= k / 2; i++)
                if (k % i == 0)
                {
                    flag = false;
                    break;
                }
            return flag;
        }
        static void Main()
        {
            int N, M, temp;
            Console.WriteLine("Введите количество строк и столбцов исходного массива");
            N = Convert.ToInt32(Console.ReadLine());
            M = Convert.ToInt32(Console.ReadLine());
            int [,] A = new int[N, M];
            int [] B = new int[N * M];
            Random rand = new Random();
            int count = 0;
            // инициализация и вывод на экран массива A
            Console.WriteLine("Массив A:");

            for (int i = 0; i < N; i++)
            {
                for (int j = 0; j < M; j++)
                {
                    A[i, j] = rand.Next(10, 99);
                    Console.Write(" {0:d2}", A[i, j]);
                }
                Console.WriteLine();
            }
            // запись в массив B элементов массива A, являющихся простыми числами
            for (int i = 0; i < N; i++)
            {
                for (int j = 0; j < M; j++)
                {
                    if (prost(A[i, j]))
                    {
                        B[count] = A[i, j];
                        count++;
                    }
                }
            }
            // сортировка массива B по возрастанию методом пузырька
            for (int i = 0; i < count-1; i++)
            {
                for (int j = i + 1; j < count; j++)
                {
                    if (B[j] < B[i])
                    {
                        temp = B[j];
                        B[j] = B[i];
                        B[i] = temp;
                    }
                }
            }
            // вывод на экран массива B
            Console.WriteLine("Массив B:");
            for (int i = 0; i < count; i++)
            {
                Console.Write(B[i] + " ");
            }
            Console.ReadKey();
        }
    }
}
```

## 1.1.5 Результаты тестирования программы 1

### Тест 1

```
Введите количество строк и столбцов исходного массива
6
8
Массив A:
 86 28 15 93 34 85 17 68
 96 28 14 48 23 50 73 73
 88 30 35 62 60 18 92 52
 51 32 33 83 40 53 50 61
 49 46 15 92 10 66 65 72
 71 44 51 87 39 31 75 26
Массив B:
17 23 31 53 61 71 73 73 83 _
```

### Тест 2

```
Введите количество строк и столбцов исходного массива
10
9
Массив A:
 29 38 58 62 30 43 50 56 44
 56 82 17 85 36 25 33 59 35
 20 85 60 84 37 58 54 41 96
 29 27 83 55 32 50 82 51 80
 78 35 35 10 32 37 36 18 85
 75 53 27 85 85 73 79 32 32
 13 44 11 10 18 88 46 30 39
 60 11 59 66 12 84 43 53 33
 98 19 62 32 57 28 71 77 86
 79 42 41 19 74 98 27 79 17
Массив B:
11 11 13 17 17 19 19 29 29 37 37 41 41 43 43 53 53 59 59 71 73 79 79 79 83 _
```

## 4.2 Пример выполнения задания 2

### 4.2.1 Постановка задачи 2

В заданном в файле `input.txt` тексте определить частоту, с которой в тексте встречаются различные буквы русского алфавита (в процентах от общего количества русских букв). Строчные и прописные буквы при этом считаются не различимыми. Результат записать в файл `output.txt` и вывести на экран.

#### 4.2.2 Таблица идентификаторов (спецификации) для задачи 2

Имя идентификатора	Описание идентификатора	Тип идентификатора
path1	Полное имя файла с исходным текстом	строка символов
path2	Полное имя выходного файла	строка символов
chastoti	Массив частот заглавных и строчных русских букв	массив целых чисел
text	Переменная для хранения исходного текста	строка символов
output_text	Переменная для хранения результата	строка символов
count	Количество русских букв тексте	целый
i	Счетчик цикла	целый

#### 4.2.3 Описание алгоритма решения задачи 2

Сначала исходный текст из файла input.txt записывается в стоковую переменную text.

Затем из текста выделяется очередной символ и проверяется, является ли он русской буквой (код символа должен быть больше либо равен кода буквы «А» и меньше либо равен кода буквы «я»).

Если символ является русской буквой, его частота увеличивается на 1. Индексом элемента массива chastoti является порядковый номер русской буквы в алфавите, начиная с нуля (разность между кодом буквы и кодом буквы «А»).

Частоты одинаковых букв (строчных и прописных) складываются.

Для каждой буквы, частота которой в тексте не равна нулю, вычисляется относительная частота в процентах и записывается в переменную output\_text.

Значение переменной output\_text записывается в файл output.txt и выводится на экран.

Если в тексте нет русских букв, то программа выдает соответствующее сообщение.

## 4.2.4 Листинг программы 2

```
using System;
using System.IO;
namespace Символьные_массивы
{
    class Program
    {
        static void Main()
        {
            string path1 = "C:\\text\\input1.txt", path2 = "C:\\text\\output.txt";
            int [] chastoti = new int[64];
            string text = File.ReadAllText(path1, System.Text.Encoding.GetEncoding(1251));
            Console.WriteLine("Исходный текст:");
            Console.WriteLine(text);
            string output_text = "";
            int count=0;
            for (int i = 0; i < text.Length; i++)
                if (((int)(text[i]) >= (int)'А') && ((int)(text[i]) <= (int)'я'))//является ли символ русской буквой
                {
                    chastoti[(int)text[i] - (int)'А']++; // подсчет частот русских букв
                    count++;
                }
            for (int i = 0; i < 32; i++)
            {
                chastoti[i] = chastoti[i] + chastoti[i + 32]; // сложение частот строчных и заглавных букв
                if (chastoti[i]!=0)
                    output_text += ((char)(i+(int)'А') + " - " + Math.Round(chastoti[i]*100.0/count)+" %"+"r\n");
            }
        }
    }

    if (output_text != "")
    {
        Console.WriteLine("Частота букв в тексте:");
        Console.WriteLine(output_text);
        File.WriteAllText(path2, output_text);
    }
    else Console.WriteLine("Русских букв в тексте нет");

    Console.ReadKey();
}
```

## 4.2.5 Результаты тестирования программы 2

### Тест 1

```
Исходный текст:
The Microsoft company employs 85,000 people. Most of them live in America.
Русских букв в тексте нет
```

## Тест 2

```
Исходный текст:  
В фирме Microsoft работают 85000 человек. Большая часть из них проживают в Америке.  
Частота букв в тексте:  
А - 11 %  
Б - 4 %  
В - 7 %  
Е - 9 %  
Ж - 2 %  
З - 2 %  
И - 9 %  
К - 4 %  
Л - 4 %  
М - 4 %  
Н - 2 %  
О - 7 %  
П - 2 %  
Р - 7 %  
С - 2 %  
Т - 7 %  
Ф - 2 %  
Х - 2 %  
Ч - 4 %  
Ш - 2 %  
Ь - 4 %  
Ю - 4 %
```

## **ЗАКЛЮЧЕНИЕ**

В методических рекомендациях к выполнению курсовой работы по дисциплине «Основы алгоритмизации и программирования» приведены варианты заданий, требования к структуре и оформлению пояснительной записки, а также примеры выполнения заданий.

Теоретические материалы, необходимые для выполнения работы, содержатся в электронном курсе лекций по дисциплине.