

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

Отчёт
к лабораторной работе
на тему

ШИФР ЦЕЗАРЯ. ШИФР ВИЖЕНЕРА

Студент: гр.153501
Савончик Е. В.

Проверил: Лещенко Е. А.

Минск 2024

СОДЕРЖАНИЕ

1 Цель работы	3
2 Проектирование алгоритма	4
3 Демонстрация программного средства	5
Приложение А Блок-схема функции шифрования с помощью шифра Цезаря	6
Приложение Б Блок-схема функции дешифрования с помощью шифра Цезаря	7
Приложение В Блок-схема функции шифрования с помощью шифра Виженера	8
Приложение Г Блок-схема функции дешифрования с помощью шифра Виженера	9
Приложение Д Исходный код программы	10

1 ЦЕЛЬ РАБОТЫ

Изучить теоретические сведения о шифре Цезаря и шифре Виженера. Реализовать программные средства шифрования и дешифрования текстовых файлов при помощи шифра Цезаря и шифра Виженера.

2 ПРОЕКТИРОВАНИЕ АЛГОРИТМА

Функции шифрования и дешифрования с помощью шифра Цезаря и шифра Виженера весьма похожи и отличаются лишь блоком выбора закодированного символа. Блок-схемы алгоритмов представлены в приложениях А-Г.

3 ДЕМОНСТРАЦИЯ ПРОГРАММНОГО СРЕДСТВА

В результате выполнения лабораторной работы было получено программное средство, способное считывать данные из файла и шифровать/дешифровать их. Весь код программы был написан на языке C#.

При запуске программы происходит считывание файла и вызов соответствующих функций классов *CaesarCipher* и *VigenereCypher*. Вывод программы после выполнения представлен на рисунках 1, 2, 3, 4.

```
Choose cipher:
1)Ceasar
2)Vigenere
1
Enter step:
5
Source: The quick brown fox jumps over the lazy dog
Encrypt: Ymj vznhp gwtbs ktc ozrux tajw ymj qfed itl
```

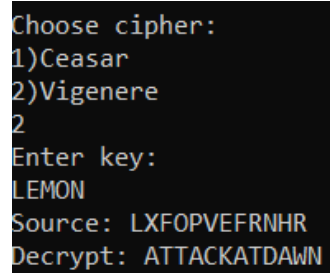
Рисунок 1 – Шифрование с помощью шифра Цезаря

```
Choose cipher:
1)Ceasar
2)Vigenere
1
Enter step:
5
Source: Ymj vznhp gwtbs ktc ozrux tajw ymj qfed itl
Decrypt: The quick brown fox jumps over the lazy dog
```

Рисунок 2 – Дешифрование с помощью Шифра Цезаря

```
Choose cipher:
1)Ceasar
2)Vigenere
2
Enter key:
LEMON
Source: ATTACKATDAWN
Encrypt: LXFOPVEFRNHR
```

Рисунок 3 – Шифрование с помощью шифра Виженера



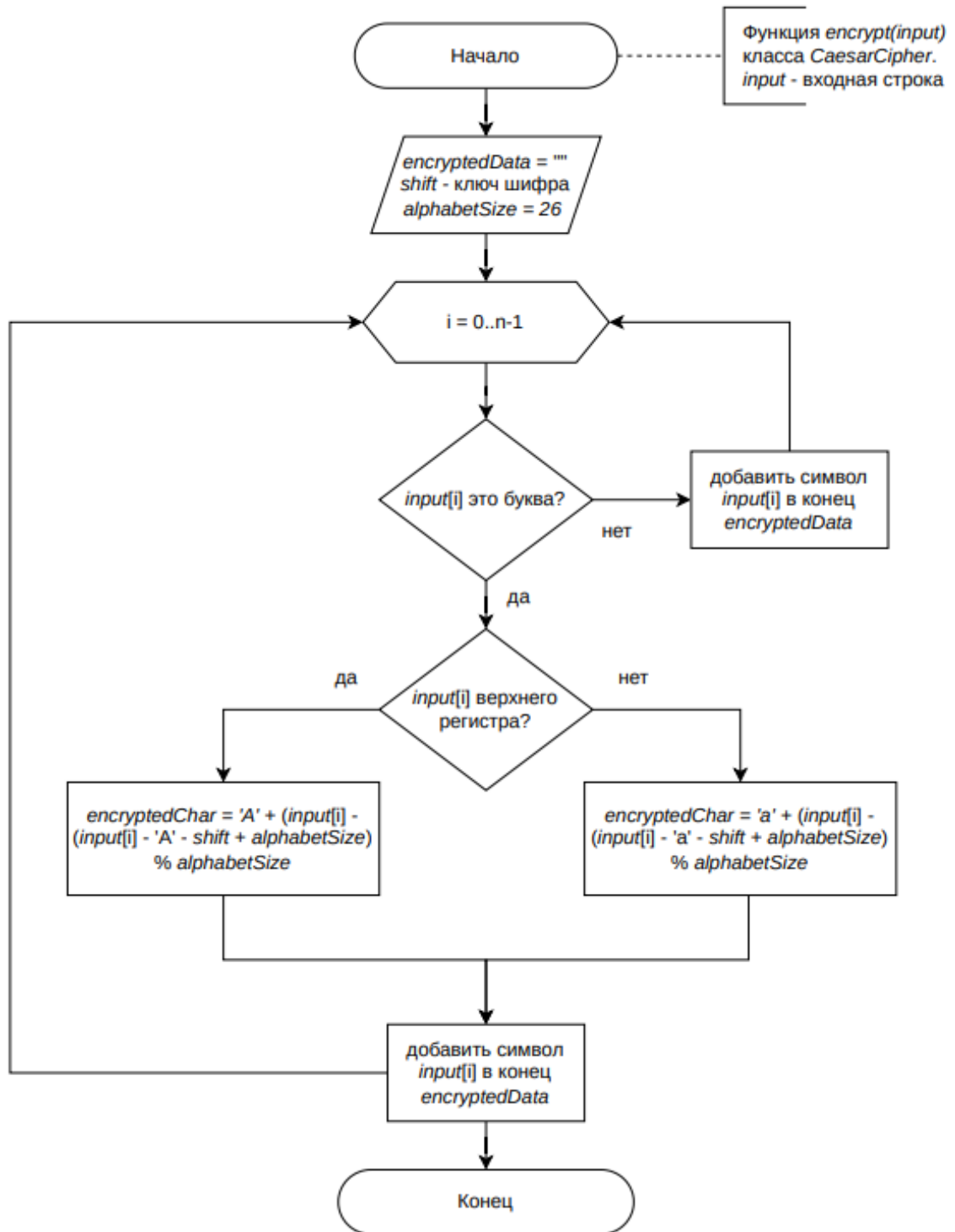
```
Choose cipher:  
1)Ceasar  
2)Vigenere  
2  
Enter key:  
LEMON  
Source: LXFOPVEFRNHR  
Decrypt: ATTACKATDAWN
```

Рисунок 1 – Дешифрование с помощью шифра Виженера

Исходный код программы представлен в приложении Д.

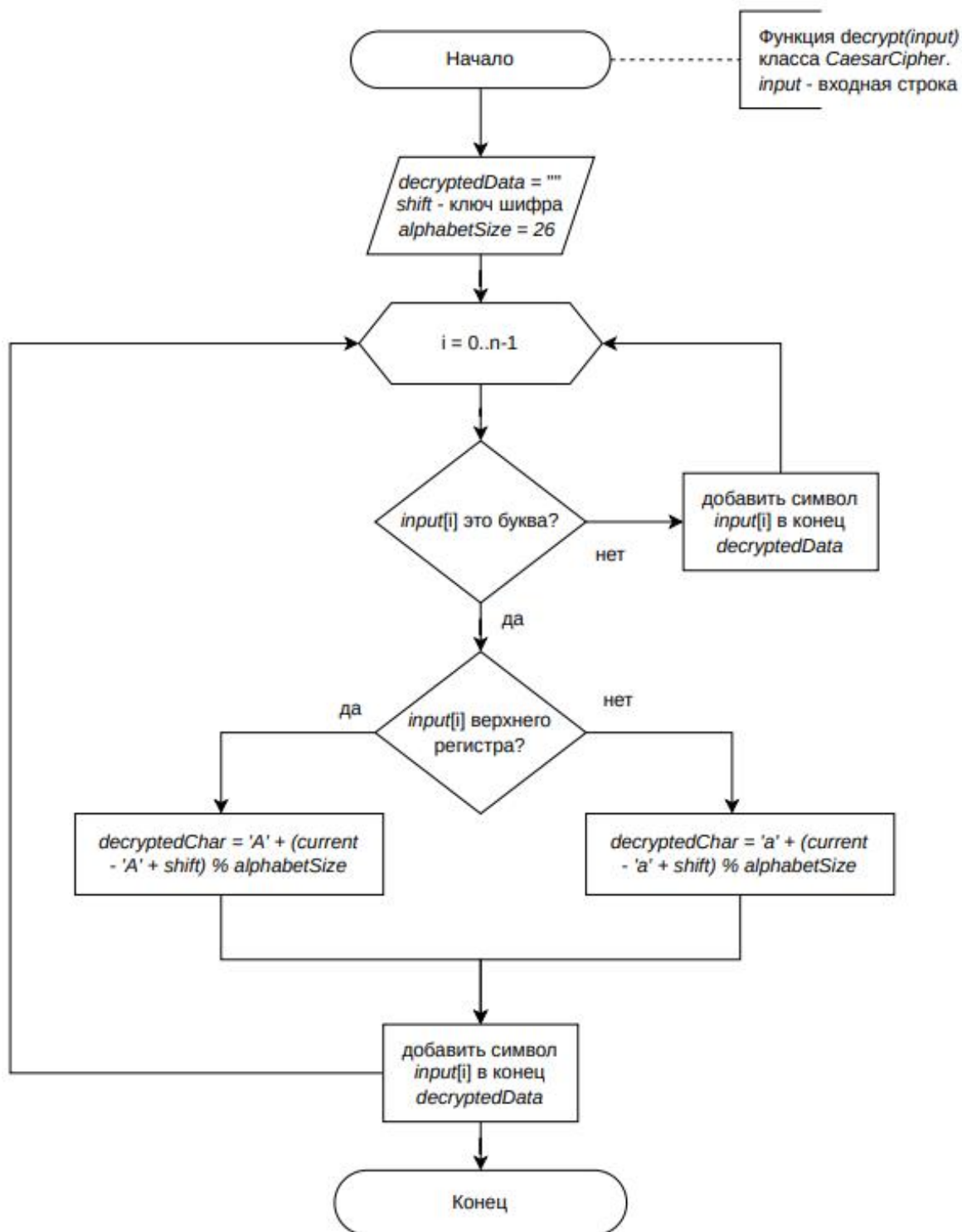
ПРИЛОЖЕНИЕ А

Блок-схема функции шифрования с помощью шифра Цезаря



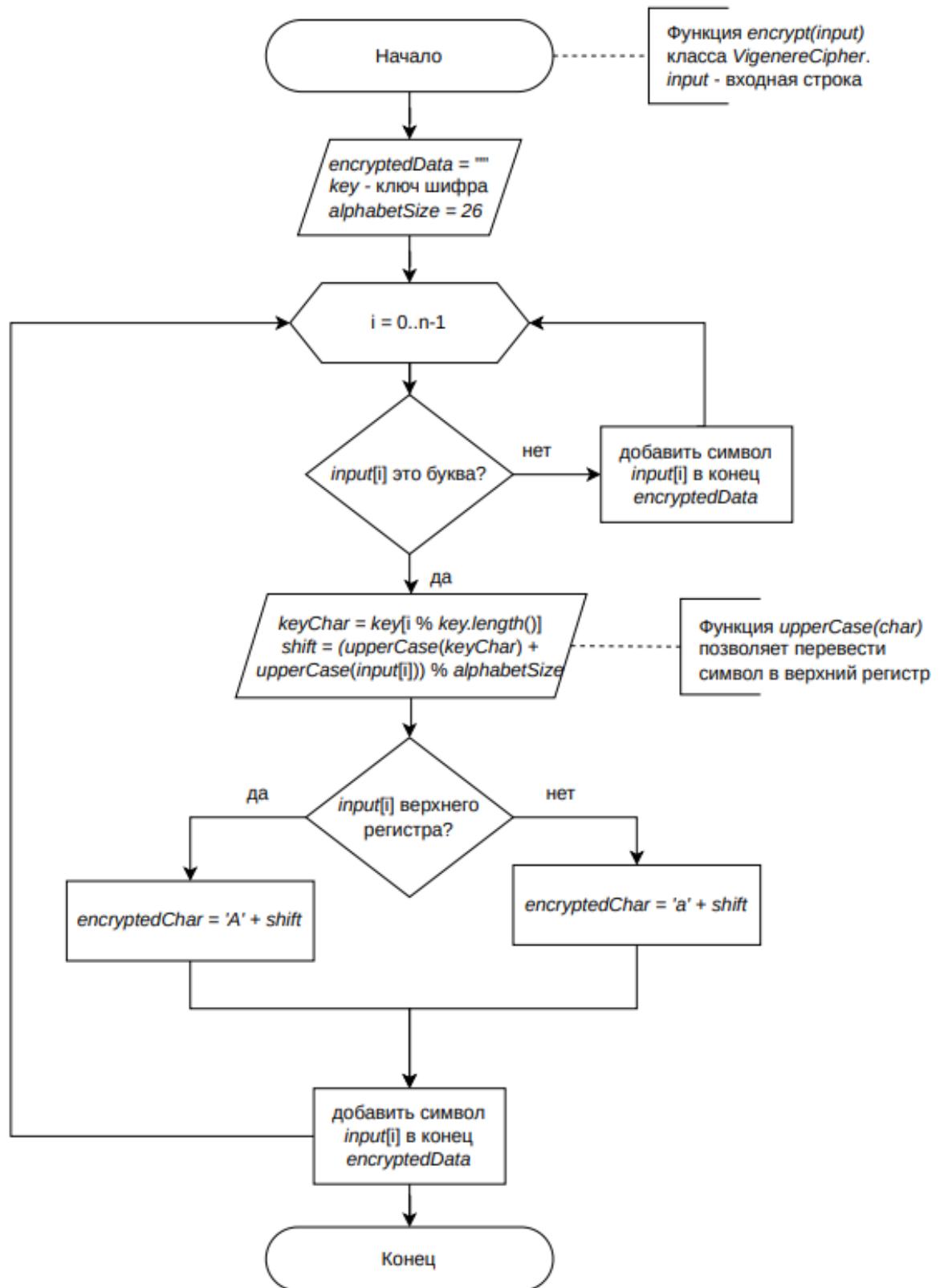
ПРИЛОЖЕНИЕ Б

Блок-схема функции дешифрования с помощью шифра Цезаря



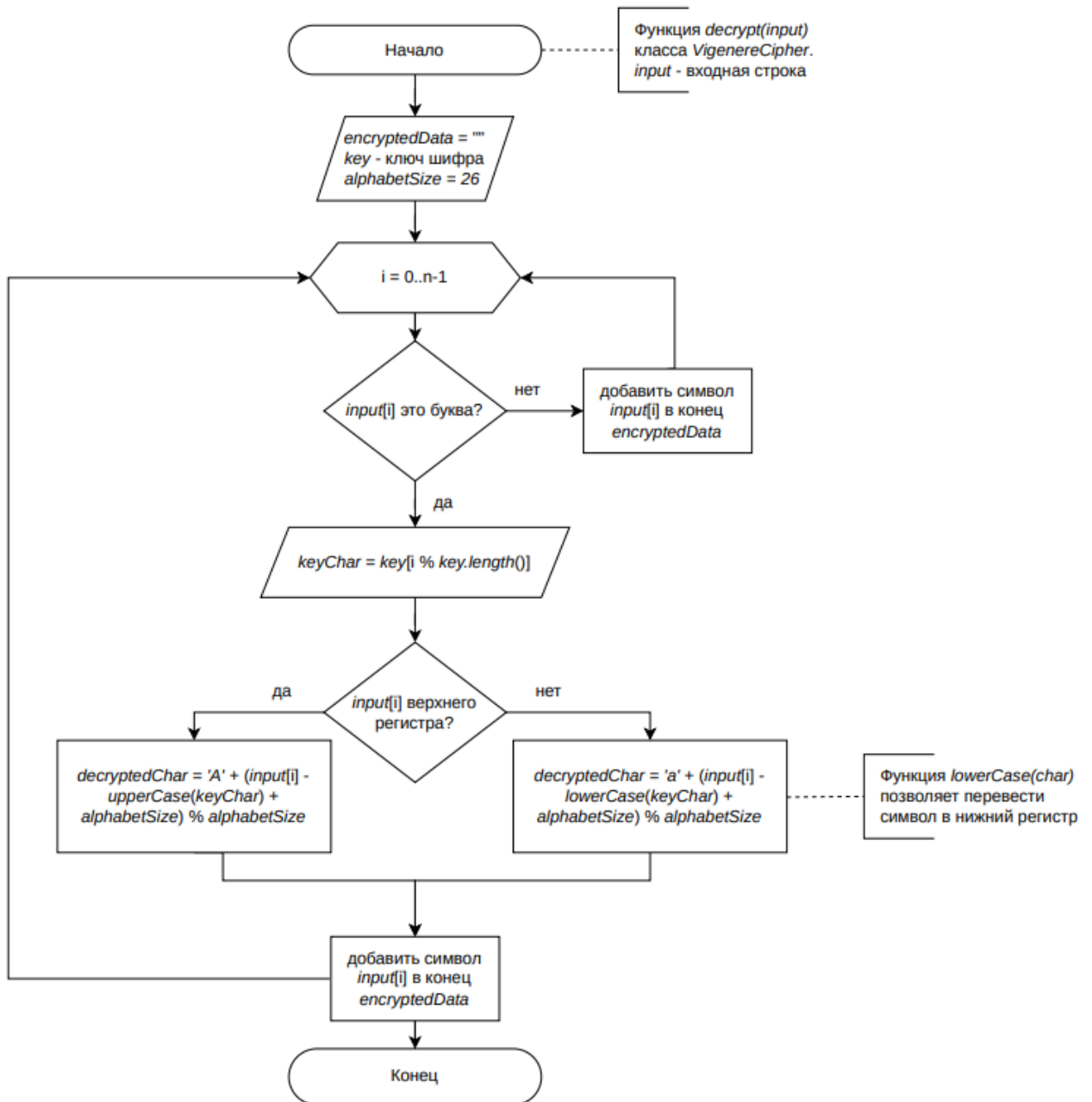
ПРИЛОЖЕНИЕ В

Блок-схема функции шифрования с помощью шифра Виженера



ПРИЛОЖЕНИЕ Г

Блок-схема функции дешифрования с помощью шифра Виженера



ПРИЛОЖЕНИЕ Д

Исходный код программы

Листинг 1 – Файл Programm.cs

```
namespace Lab_1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Test filename:
C:\\Users\\HP\\Desktop\\labs\\LABS_BSUIR\\6sem_ISOБ_C#\\Labs_ISOБ\\Lab_1\\Tes
t\\CDecryptTest.txt");

            Console.WriteLine("Enter filename: ");
            string? fileName = Console.ReadLine();
            string source = "";

            if (File.Exists(fileName))
            {
                try
                {
                    source = File.ReadAllText(fileName);
                }
                catch (IOException e)
                {
                    Console.WriteLine($"Reading file error: {e.Message}");
                }
            }
            else
            {
                Console.WriteLine("File not exists");
                return;
            }
        }
    }
}
```

```

Console.WriteLine("Choose action:\n1)Encrypt\n2)Decrypt");
string choiceAction = Console.ReadLine();

Console.WriteLine("Choose cipher:\n1)Ceasar\n2)Vigenere");
string choiceCipher = Console.ReadLine();

if (choiceCipher == "1")
{
    Console.WriteLine("Enter step: ");

    CaesarCipher cc = new
CaesarCipher(Convert.ToInt32(Console.ReadLine()));

    Console.WriteLine("Source: " + source);
    if(choiceAction == "1")
    {
        Console.WriteLine("Encrypt: " + cc.Encrypt(source));
    }
    else if(choiceAction == "2")
    {
        Console.WriteLine("Decrypt: " + cc.Decrypt(source));
    }
}
else if (choiceCipher == "2")
{
    Console.WriteLine("Enter key: ");

    VigenereCipher vc = new
VigenereCipher(Console.ReadLine().Trim());

    Console.WriteLine("Source: " + source);
    if (choiceAction == "1")
    {
        Console.WriteLine("Encrypt: " + vc.Encrypt(source));
    }
    else if (choiceAction == "2")
    {
        Console.WriteLine("Decrypt: " + vc.Decrypt(source));
    }
}

```

```

        }
    }
    else
    {
        Console.WriteLine("Invalid value");
    }
}
}
}
}

```

Листинг 2 – Файл CaesarCipher.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab_1
{
    public class CaesarCipher
    {
        public int Step { get; set; }
        private const int ALPHABET_LENGTH = 26;

        public CaesarCipher(int step)
        {
            Step = step;
        }

        public string Encrypt(string source)
        {
            StringBuilder result = new StringBuilder();

```

```

        foreach (char letter in source)
        {
            if(char.IsLetter(letter) && char.IsUpper(letter))
            {
                result.Append((char)((letter + Step - 'A') %
ALPHABET_LENGTH + 'A'));
            }
            else if(char.IsLetter(letter))
            {
                result.Append((char)((letter + Step - 'a') %
ALPHABET_LENGTH + 'a'));
            }
            else
            {
                result.Append(letter);
            }
        }

        return result.ToString();
    }

    public string Decrypt(string source)
    {
        StringBuilder result = new StringBuilder();

        foreach (char letter in source)
        {
            if (char.IsLetter(letter) && char.IsUpper(letter))
            {
                result.Append((char)((letter - Step + ALPHABET_LENGTH -
'A') % ALPHABET_LENGTH + 'A'));
            }
            else if (char.IsLetter(letter))
            {
                result.Append((char)((letter - Step + ALPHABET_LENGTH -
'a') % ALPHABET_LENGTH + 'a'));
            }
        }
    }
}

```

```

        }
        else
        {
            result.Append(letter);
        }
    }

    return result.ToString();
}
}
}

```

Листинг 3 – Файл VigenereCipher.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab_1
{
    public class VigenereCipher
    {
        private string _key;

        public string Key
        {
            get => _key;
            set
            {
                if (value != null)
                    _key = value.ToLower();
            }
        }
    }
}

```

```

    }

    private const int ALPHABET_LENGTH = 26;

    public VigenereCipher(string key)
    {
        Key = key;
    }

    public string Encrypt(string source)
    {
        StringBuilder result = new StringBuilder();

        for (int i = 0; i < source.Length; i++)
        {
            int letOffset = char.IsUpper(source[i]) ? 'A' : 'a';

            if (char.IsLetter(source[i]))
            {
                result.Append((char) (((source[i] + Key[i % Key.Length] -
letOffset - 'a') % ALPHABET_LENGTH) + letOffset));
            }
        }

        return result.ToString();
    }

    public string Decrypt(string source)
    {
        StringBuilder result = new StringBuilder();

        for (int i = 0; i < source.Length; i++)
        {
            int letOffset = char.IsUpper(source[i]) ? 'A' : 'a';

```



```

        if (char.IsLetter(source[i]))
        {
            result.Append((char)((source[i] - Key[i % Key.Length] +
ALPHABET_LENGTH - letOffset + 'a') % ALPHABET_LENGTH) + letOffset));
        }
    }

    return result.ToString();
}
}
}

```