Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«КУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №2**

«Программирование на языке C с использованием нуль-терминальных строк»

дисциплина «Основы алгоритмизации и программирования»

Выполнил:

студент группы 113 Гришанов Е.А.

Проверил:

старший преподаватель кафедры ПОиАИС Ураева Е.Е.

Курск, 2024

***Цель работы:*** изучить особенности написания программ на языке С с использованием нуль-терминальных строк.

***Задание***

*Задача 1.* Даны две строки str1 и str2. Получить новую строку, в которой записаны все символы строки str1 до последнего вхождения строки str2 в str1.

*Задача 2.* Дан текст. Определить частоту появления букв "а" в нем. Частота вычисляется как отношение количества данных символов в тексте к длине всего текста (пробелы учитываются, а символ конца строки не учитывается).

*Задача 3.* Написать функцию, формирующую строку из исходной, используя шифр Цезаря (каждая буква заменяется на следующую по алфавиту через K позиций по кругу). Функция должна иметь прототип: char\* str(char\* str, int k), где str – строка-источник, k – код шифра. При определении функции задачи 3 недопустимо использовать строковые функции и операцию индексации. Обеспечить возможность ввода пользователем данных, вызов описанной функции и вывод результата ее работы.

***Разработка алгоритма***

*Задача 1*

Входные данные алгоритма решения задачи:

* *str1, str2* – нуль-терминальные строки.

Выходные данные алгоритма решения задачи:

* *str3* – нуль-терминальная строка.

Алгоритм решения задачи представлен на рисунке 1.

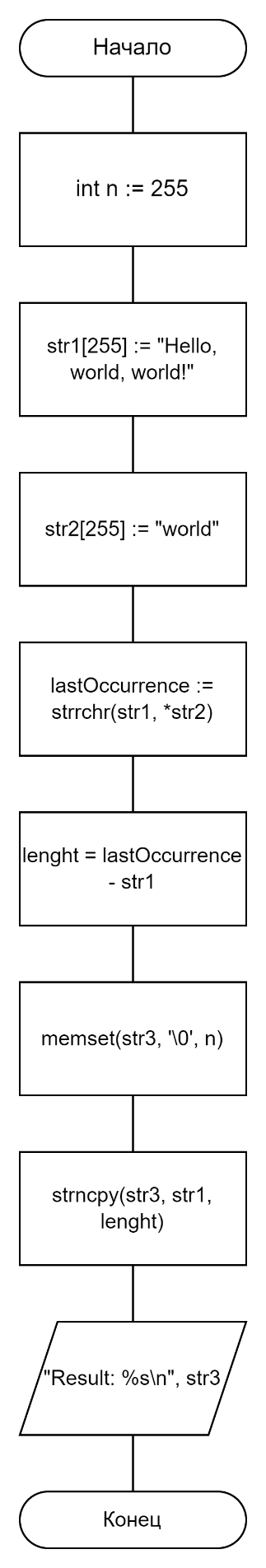


Рисунок 1 – Алгоритм решения задачи 1

*Задача 2*

Входные данные алгоритма решения задачи:

* *str –* нуль-терминальная строка;

Выходные данные алгоритма решения задачи:

* *s –* нуль-терминальная строка.

Алгоритм решения задачи представлен на рисунке 2.

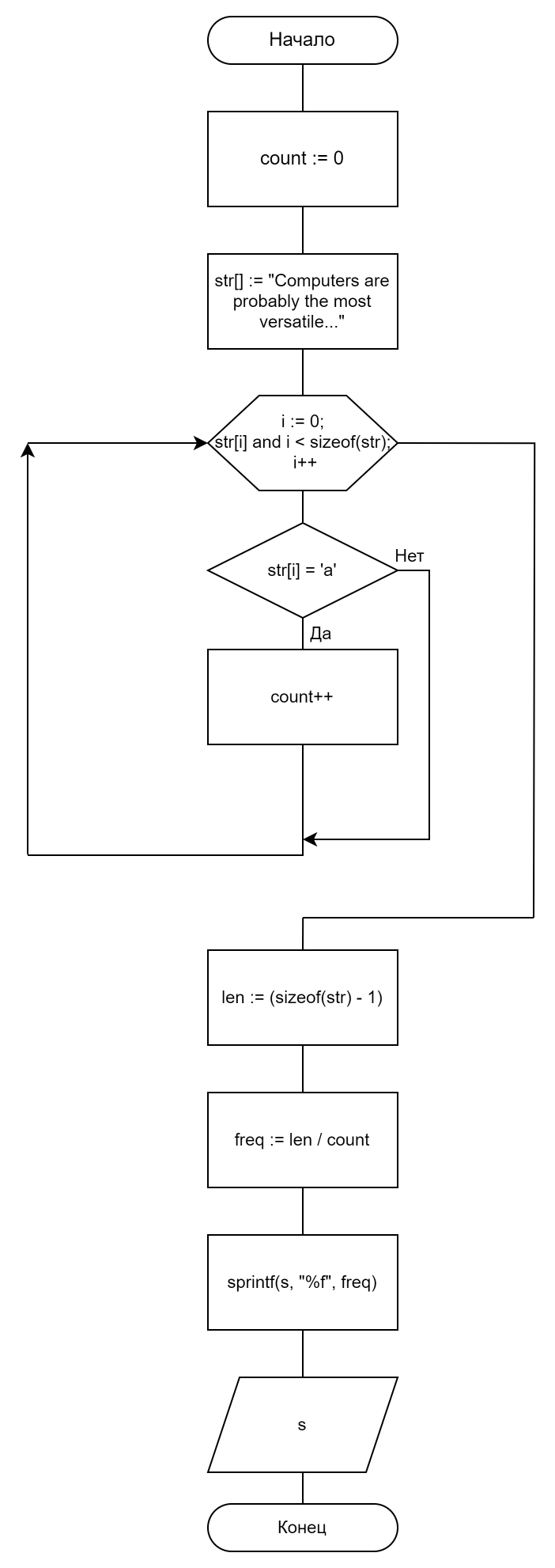


Рисунок 2 – Алгоритм решения задачи 2

*Задача 3*

Основной алгоритм решения задачи реализован с использованием следующей функции:

*char\* str(char\* str, int k)*

Функция str выполняет основную функцию программы – кодирует текст согласно заданию.

Входные аргументы:

- str – указатель на строку,

- k – целое число.

Выходные аргументы:

*-* str – нуль-терминальная строка.

Входные данные алгоритма решения задачи:

* input – нуль-терминальная строка;
* k – целое число.

Выходные данные алгоритма решения задачи:

* *res* – нуль-терминальная строка.

Алгоритм решения задачи представлен на рисунках 3, 4.

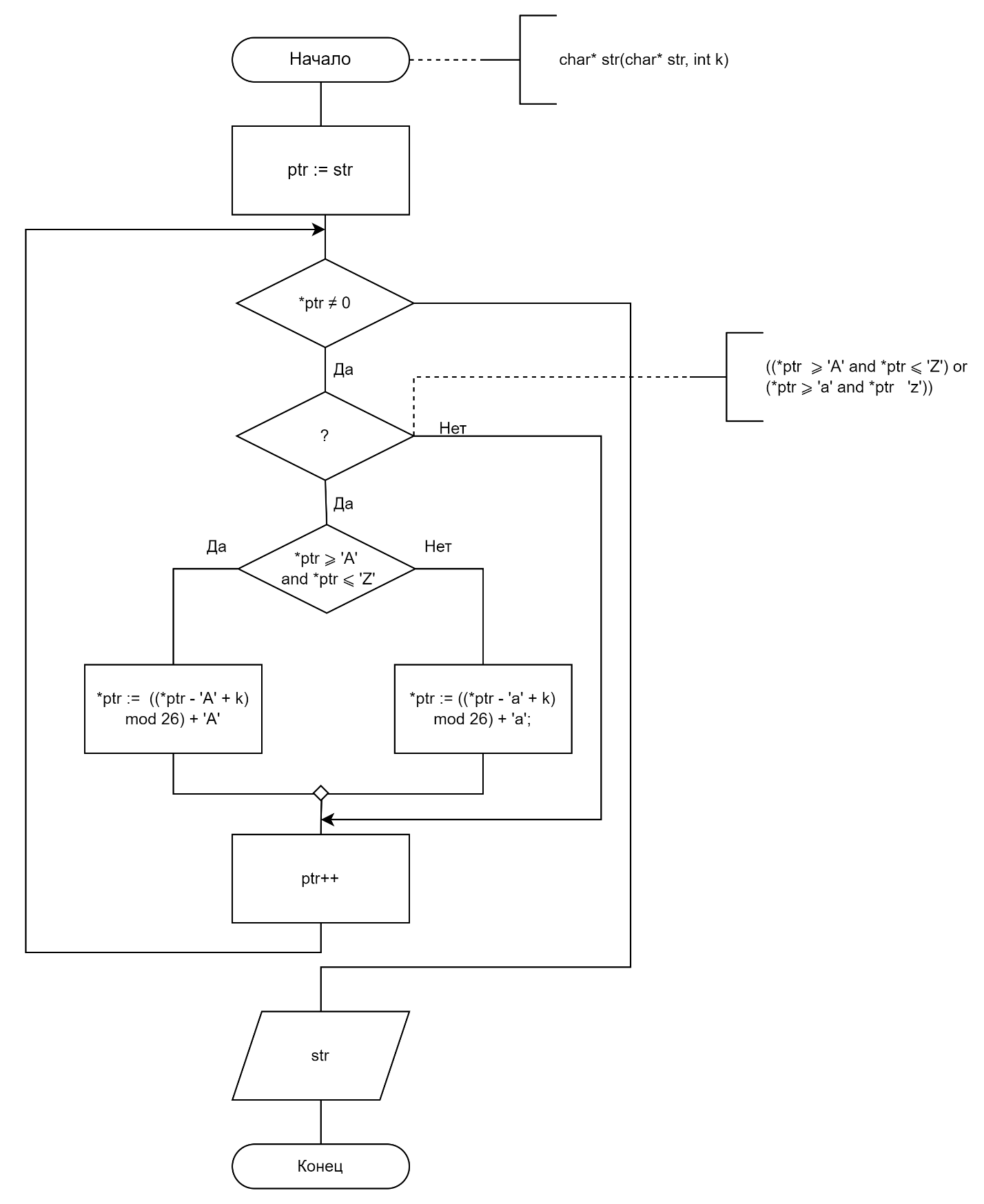


Рисунок 3 – Алгоритм функции str.

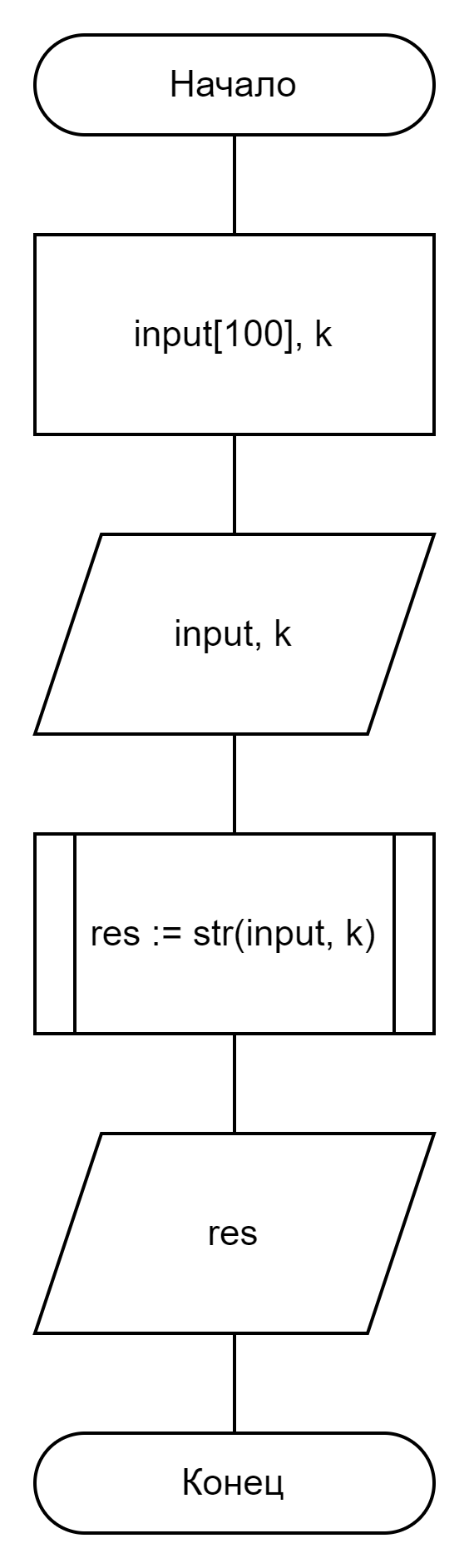


Рисунок 4 – Алгоритм решения задачи.

***Текст программы***

*Текст программы для решения задачи 1*

#include <stdio.h>

#include <string.h>

int main(){

const int n = 255;

// Создаем строки

char str1[255] = "Hello, world, world!";

char str2[255] = "world";

//Находим последнее вхождение str2 в str1

char\* lastOccurrence = strrchr(str1, \*str2);

// Выделяем память под новую строку

int lenght = lastOccurrence - str1;

char str3[n];

memset(str3, '\0', n);

// Копируем в новую строку все символы str1 до последнего вхождения str2

strncpy(str3, str1, lenght);

// Выводим результат

printf("Result: %s\n", str3);

}

*Текст программы для решения задачи 2*

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <stdlib.h>

using namespace std;

int main(){

double count = 0;

char s[10];

char str[] = "Computers are probably the most versatile tools that humanity has at its disposal. They are capable of performing incredible calculations, they allow you to store a huge amount of information, completely in different parts of the planet, and at the same time easily exchange it, regardless of location.";

for(int i = 0; str[i] && i < sizeof(str); i++){

if(str[i] == 'a')

count++;

}

double len = (sizeof(str) - 1);

double freq = len / count;

sprintf(s, "%f", freq);

printf(s);

}

*Текст программы для решения задачи 3*

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

using namespace std;

char\* str(char\* str, int k){

char\* ptr = str;

while (\*ptr != 0){

if((\*ptr >= 'A' && \*ptr <= 'Z') || (\*ptr >= 'a' && \*ptr <= 'z')){

if(\*ptr >= 'A' && \*ptr <= 'Z'){

\*ptr = ((\*ptr - 'A' + k) % 26) + 'A';

}

else{

\*ptr = ((\*ptr - 'a' + k) % 26) + 'a';

}

}

ptr++;

}

return str;

}

int main(){

char input[100];

int k;

printf("Enter string: ");

scanf("%s", input);

printf("Enter the code of cipher: ");

scanf("%d", &k);

char\* res = str(input, k);

printf(res);

}

***Тестирование программы***

Тестирование задачи 1 представлено на рисунке 5.



Рисунок 5 –Тест 1 задачи 1

Тестирование задачи 2 представлено на рисунке 6.



Рисунок 6 –Тест 1 задачи 2

Тестирование задачи 3 представлено на рисунках 7, 8.



Рисунок 7 – Тест 1 Задачи 3



Рисунок 8 – Тест 2 Задачи 3