МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ образовательное учреждениевысшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

**КОНТРОЛЬНАЯ РАБОТА**

**по дисциплине «ИНФОРМАТИКА»**

|  |  |
| --- | --- |
| Факультет: АВТ  Группа: ДТ-460а  Студент: Дроздов И.С. | Преподаватель:  Копылова О.А. |

Новосибирск, 2025 г.

Содержание

[ЗАДАНИЕ №1. Анализ программ 3](#_Toc199336064)

[ЗАДАНИЕ №2. Строки 4](#_Toc199336065)

[ЗАДАНИЕ №3. Итерационные циклы 4](#_Toc199336066)

[Выводы 4](#_Toc199336067)

# ЗАДАНИЕ №1. Анализ программ

Данная функция разворачивает массив в обратном порядке от изначального используя сортировку пузырьком. После объявления переменных следует цикл, в условии которого идёт перебор массива в обратном порядке, а после в теле цикла, элементы сортируются в обратном порядке.

Вариант задания 18:

void F18(int c[], int n)

{

int i,j,k;

for (i=0,j=n-1; i < j; i++,j--){

k = c[i];

c[i] = c[j];

c[j] = k;

}

}

**Решение**

Исходный код программы с комментариями и вызовом функции:

//---------------------------------------------------Задание1. Фрагмент 18.

#include <stdio.h>

int F18\_call(){

// Пример 1: Разворот массива из 5 элементов

int arr1[] = {1, 2, 3, 4, 5};

F18(arr1, 5);

printf("Пример 1: ");

for (int i = 0; i < 5; i++) printf("%d ", arr1[i]); // Выведет: 5 4 3 2 1

// Пример 2: Разворот массива из 6 элементов (чётное количество)

int arr2[] = {10, 20, 30, 40, 50, 60};

F18(arr2, 6);

printf("\nПример 2: ");

for (int i = 0; i < 6; i++) printf("%d ", arr2[i]); // Выведет: 60 50 40 30 20 10

printf("\n");

return 0;

}

void main(){

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Задача №1 перестановка элементов массива");

F18\_call();

}

Пояснения к коду:

Функция возвращает переданный в неё массив чисел в обратном порядке от исходного.

**Пример работы программы**

Пример работы программы представлен на рисунке:



Рисунок 1 – Пример работы программы

# ЗАДАНИЕ №2. Строки

Данная функция ищет наибольшее повторение заглавных букв в строке и выводит ту, которая чаще всего встречается.

Вариант задания 10:

char F10(char c[]){

char m,z;

int n,s,i;

for (s=0,m='A'; m <='Z'; m++){

for(n=0, i=0; c[i] !='\0'; i++)

if (c[i]==m) n++;

if (n > s) { z=m; s=n; }

}

return z;

}

**Решение**

Исходный код программы с комментариями и вызовом функции:

//---------------------------------------------------Задание2. Фрагмент 10.

#include <stdio.h>

int F10\_call() {

// Пример 1: Строка с одной самой частой буквой

char str1[] = "HELLO WORLD";

char result1 = F10(str1);

printf("\nПример 1: %c\n", result1);

// Пример 2: Строка с несколькими буквами одинаковой частоты

char str2[] = "ABBCCCDDDD";

char result2 = F10(str2);

printf("Пример 2: %c\n", result2);

// Пример 3: Пустая строка

char str3[] = "";

char result3 = F10(str3);

printf("Пример 3: %c (для пустой строки)\n", result3);

return 0;

}

void main(){

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Задача №2: строки\n");

F10\_call();

}

Пояснения к коду:

Функция возвращает самую частовстречаемую заглавную букву в строке.

**Пример работы программы**

Пример работы программы представлен на рисунке:

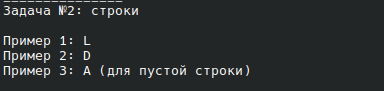


Рисунок 2 – Пример работы программы

# ЗАДАНИЕ №3. Итерационные циклы

Данная функция реализовывает вычисление гиперболического синуса при помощи разложения ряда Тейлора, вычисляя приближённое значение ряда суммируя первые n\_term членов. Данная реализация вычисление гиперболического синуса лишена нахождения факториала и возведения в степень и основана только на рекурсивных операциях с умножением и делением.

Вариант задания 10:

double sh\_taylor\_optimized(double x, int n\_terms) {

double sum = x;

double term = x;

for (int n = 2; n <= n\_terms; ++n) {

term \*= x \* x / ((2 \* n - 2) \* (2 \* n - 1));

sum += term;

}

return sum;

}

**Решение**

Исходный код программы с комментариями и вызовом функции:

//---------------------------------------------------Задание3. Фрагмент 5.

#include <stdio.h>

#include <math.h>

void sh\_taylor\_test() {

const double x\_start = 0.1;

const double x\_end = 1.0;

const double x\_step = 0.1;

const int terms = 10;

printf(" x | Taylor (2) | Taylor (4) | Taylor (5) | Math lib | Term -> 0?\n");

printf("-----|------------|------------|------------|------------|------------\n");

for (double x = x\_start; x <= x\_end + 1e-9; x += x\_step) {

double taylor = sh\_taylor\_optimized(x, terms);

double exact = sinh(x);

double last\_term = x;

for (int n = 2; n <= terms; n++) {

last\_term \*= x \* x / ((2 \* n - 2) \* (2 \* n - 1));

}

printf("%.1f | %-10.2f | %-10.4f | %-10.5f | %-10.6f | %s\n",

x, taylor, taylor, taylor, exact,

fabs(last\_term) < 1e-6 ? "Yes" : "No");

}

}

void main(){

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Задача №3: итерационные циклы\n");

sh\_taylor\_test();

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

Пояснения к коду:

Функция возвращает переданный в неё массив чисел в обратном порядке от исходного.

**Пример работы программы**

Пример работы программы представлен на рисунке:

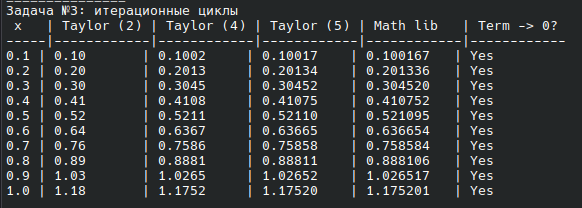


Рисунок 3 – Пример работы программы

# Выводы

В ходе выполнения контрольной работы мной были написаны программы по реализации переворачивания массива, нахождению повторяющихся букв в строке и поиск гиперболического синуса через разложение ряда Тейлора. Были получены навыки работы со строками, а так же по расписыванию формулы через рекурсию.