

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ образовательное
учреждение высшего образования**
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра вычислительной техники

КОНТРОЛЬНАЯ РАБОТА
по дисциплине «ИНФОРМАТИКА»

Факультет: АВТ

Группа: ДТ-460а

Студент: Пантюхин А.Е.

Преподаватель:

Копылова О.А.

Новосибирск, 2025 г.

Содержание

ЗАДАНИЕ №1. Анализ программ.....	3
Решение	3
Пояснения к коду:.....	4
Пример работы программы	4
ЗАДАНИЕ №2. Строки.....	4
Решение	5
Пояснения к коду:.....	5
Пример работы программы	6
ЗАДАНИЕ №3. Итерационные циклы.....	6
Решение:	6
Пример работы программы	8
Выводы.....	8

ЗАДАНИЕ №1. Анализ программ

Содержательно сформулировать результат выполнения функции, определить «смысл» отдельных переменных, найти стандартные контексты, их определяющие, написать вызов функции.

В тексте программы в комментариях указать назначение основных переменных. Для функций в комментариях указать их назначение, а также назначения входных и выходного параметров.

Вариант задания 8:

```
int F8(int a) {  
    int n,k,s;  
    for (n=a, s=0; n!=0; n=n/10)  
    { k=n%10; s=s+k;}  
    return s;
```

Решение

Исходный код программы с комментариями и вызовом функции:

```
#include <stdio.h>  
  
int F8(int a) { //на вход принимаем целое число  
    int n,k,s; //объявляем три переменных, n - бесполезная  
    //k тоже бесполезная, но чуть менее.  
    /*  
    делим число нацело на 10, пока оно не станет равным нулю  
    */  
    for (n = a,s=0; n!=0;n=n/10) {  
        k = n%10; //на каждой итерации деления записываем остаток в k  
        s=s+k; //прибавляя к s - в итоге получим сумму остатков от деления на 10.  
    }  
    //возвращаем сумму остатков  
    return s;  
}  
  
int main(void) {  
    fprintf(stdout, "F8(18745)=%d\n", F8(18745));  
    fprintf(stdout, "F8(-648)=%d\n", F8(-648));  
    return 0;
```

}

Пояснения к коду:

Функция возвращает сумму цифр полученного на вход числа.

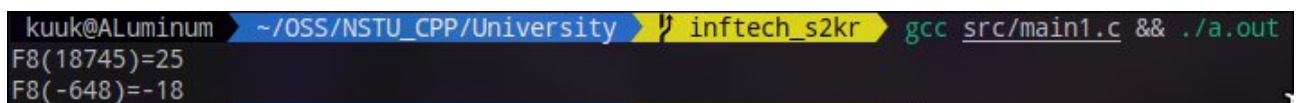
Переменная `n` бесполезна по причине того, что переменная `a` – уже копия исходного значения, и мы могли использовать её безо всяких проблем.

Аналогично, вычисление остатка можно было сократить до `s+=n%10`.

Данная функция при работе с отрицательными числами выдаст отрицательную же сумму цифр.

Пример работы программы

Пример работы программы представлен на рисунке:



```
kuuk@ALuminum ~/OSS/NSTU_CPP/University$ inftech_s2kr gcc src/main1.c && ./a.out
F8(18745)=25
F8(-648)=-18
```

Рисунок 1 – Пример работы программы

ЗАДАНИЕ №2. Строки

Содержательно определите действие, производимое над строкой.

Напишите вызов функции (входные неизменяемые строки могут быть представлены фактическими параметрами – строковыми константами). В тексте программы в комментариях указать назначение основных переменных. Для функций в комментариях указать их назначение, а также назначения входных и выходного параметров.

Вариант задания 7:

```
int F7(char c[])
{
    int i,s;
    for (i=0; c[i] !='\0'; i++)
        if (c[i] >='0' && c[i]<='7') break;
    for (s=0; c[i] >='0' && c[i] <='7'; i++)
        s = s * 8 + c[i] - '0';
    return s;
}
```

Решение

```
#include <stdio.h>

int F7(char c[]) { //на вход получаем строку
    int i,s; //объявляем счётчики
    /*
    движемся по строке, пока не встретим
    цифру в восьмеричной системе счисления
    */
    for (i=0; c[i] !='\0'; i++)
        if (c[i] >='0' && c[i]<='7') break;
    /*
    с места встречи считаем данный участок строки
    числом в восьмеричной с.с., до тех пор, пока
    не встретим неподходящий символ
    */
    for (s=0; c[i] >='0' && c[i] <='7'; i++)
        //в s формируется равное число в десятичной с.с.
        s = s * 8 + c[i] - '0';
    //возвращаем число в десятичной с.с.
    return s;
}

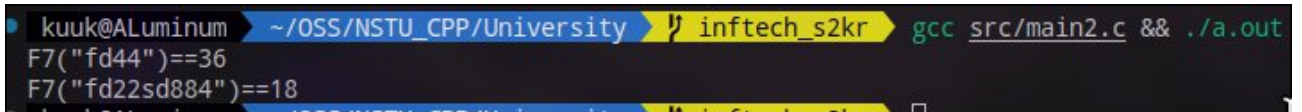
int main(void) {
    printf("F7(\"fd44\")==%d\n", F7("fd44"));
    printf("F7(\"fd22sd884\")==%d\n", F7("fd22sd884"));
    return 0;
}
```

Пояснения к коду:

Функция возвращает число в десятичной системе счисления, равное первому найденному во входной строке числу в восьмеричной системе счисления.

Функция воспринимает любое число как положительное.

Пример работы программы



```
kuuk@ALuminum ~/OSS/NSTU_CPP/University$ inftech_s2kr gcc src/main2.c && ./a.out
F7("fd44")==36
F7("fd22sd884")==18
```

Рисунок 2 – Пример работы программы

ЗАДАНИЕ №3. Итерационные циклы

Для заданного варианта написать функцию вычисления суммы ряда. Стандартный диапазон значений $x = 0.1 \dots 1.0$, шаг 0.1. Убедиться, что на этом диапазоне элемент суммы ряда стремится к 0, при необходимости скорректировать диапазон. Вычислить значения суммы ряда и контрольной функции, к которой он сходится, с точностью до 2, 4 и 5 знаков после запятой.

В тексте программы в комментариях указать назначение основных переменных. Для функций в комментариях указать их назначение, а также назначения входных и выходного параметров.

Вариант задания 4:

Функция: $\ln(x)$

Ряд: $(x-1)/x + (x-1)^2/2x^2 + (x-1)^3/3x^3 + \dots + (x-1)^n/nx^n$

Решение:

```
#include <stdio.h>
#include <math.h>
#include <locale.h>

double getValue(double x, double n) {
    return (double)(pow(x-1.0f, n)/(n*pow(x, n))); //вычисляем значение n-го члена ряда
}

double sumSequence(double x, double epsilon) {
    double result = 0.0f; //изначально сумма равна нулю
    int idx = 1; //ряд начинается с первого элемента, n==1
    double elem = getValue(x, idx); //вычисляем первый элемент

    while (fabs(elem) > epsilon) { //пока n-й элемент по модулю меньше нашего "эпсилон"
        result += elem; //суммируем элементы
```

```

        elem = getValue(x, (double)++idx); //вычисляем следующий (++idx обеспечивает
увеличение счётчика ДО копирования значения в функцию)
    }
    return result; //возвращаем результат
}

int main(void) {
    setlocale(LC_ALL, "RUSSIAN"); //установили локаль
    const double startX = 0.5f; //начало диапазона значений X
    const double increment = 0.1f; //приращение X

    const double target_epsilon[3] = {1e-2, 1e-4, 1e-5}; //целевые значения точности
    double referenceResult[3][10] = {}; //массив под результаты ln(x)
    double sumSequenceResult[3][10] = {}; //массив под результаты sumSequence(x, epsilon)
    for (int i = 0; i < 3; i++) { //вычисляем сразу три точности
        for (int j = 0; j < 10; j++) { // для каждой - по 10 экземпляров значения функции и суммы
ряда
            referenceResult[i][j] = log(startX+increment*j); //несмотря на то, что i,j - int, их можно
так использовать
            sumSequenceResult[i][j] = sumSequence(startX+increment*j, target_epsilon[i]);
        }
    }

    /*
Форматированный вывод
*/
    printf("Точность сходимости: %.e\t\t\t", target_epsilon[0]);
    printf("Точность сходимости: %.e\t\t\t", target_epsilon[1]);
    printf("Точность сходимости: %.e\n", target_epsilon[2]);
    printf("%8s\t%10s\t%8s\t", "X", "Сумма ряда", "ln(x)");
    printf("%8s\t%10s\t%8s\t", "X", "Сумма ряда", "ln(x)");
    printf("%8s\t%10s\t%8s\n", "X", "Сумма ряда", "ln(x)");
    for (int row = 0; row < 10; row++) { //выводим значения по рядам.
        for (int i = 0; i < 3; i++) {
            printf("%8.1lf\t%10.5lf\t%8.5lf\t", startX+0.1f*(double)(row), sumSequenceResult[i][row],
referenceResult[i][row]);
        }
    }
}

```

```

printf("\n");
}
return 0;
}

```

Получение значения n-го элемента ряда было решено вынести в отдельную функцию для улучшения читабельности кода. Было обнаружено, что стандартный диапазон значений показывает расходимость на промежутке [0.1;0.4], диапазон скорректирован до [0.5;1.4].

Пример работы программы

Точность сходимости: 1e-02			Точность сходимости: 1e-04			Точность сходимости: 1e-05		
X	Сумма ряда	ln(x)	X	Сумма ряда	ln(x)	X	Сумма ряда	ln(x)
0.1	-nan	-2.30259	0.1	-nan	-2.30259	0.1	-nan	-2.30259
0.2	-nan	-1.60944	0.2	-nan	-1.60944	0.2	-nan	-1.60944
0.3	-nan	-1.20397	0.3	-nan	-1.20397	0.3	-nan	-1.20397
0.4	-nan	-0.91629	0.4	-nan	-0.91629	0.4	-nan	-0.91629
0.5	-0.69817	-0.69315	0.5	-0.69268	-0.69315	0.5	-0.69268	-0.69315
0.6	-0.50553	-0.51083	0.6	-0.51088	-0.51083	0.6	-0.51082	-0.51083
0.7	-0.36297	-0.35667	0.7	-0.35664	-0.35667	0.7	-0.35667	-0.35667
0.8	-0.21875	-0.22314	0.8	-0.22318	-0.22314	0.8	-0.22314	-0.22314
0.9	-0.11111	-0.10536	0.9	-0.10540	-0.10536	0.9	-0.10536	-0.10536
1.0	0.00000	0.00000	1.0	0.00000	0.00000	1.0	0.00000	0.00000

Рисунок 3 – результаты без корректировки диапазона.

Точность сходимости: 1e-02			Точность сходимости: 1e-04			Точность сходимости: 1e-05		
X	Сумма ряда	ln(x)	X	Сумма ряда	ln(x)	X	Сумма ряда	ln(x)
0.5	-0.69817	-0.69315	0.5	-0.69268	-0.69315	0.5	-0.69268	-0.69315
0.6	-0.50553	-0.51083	0.6	-0.51088	-0.51083	0.6	-0.51082	-0.51083
0.7	-0.36297	-0.35667	0.7	-0.35664	-0.35667	0.7	-0.35667	-0.35667
0.8	-0.21875	-0.22314	0.8	-0.22318	-0.22314	0.8	-0.22314	-0.22314
0.9	-0.11111	-0.10536	0.9	-0.10540	-0.10536	0.9	-0.10536	-0.10536
1.0	0.00000	0.00000	1.0	0.00000	0.00000	1.0	0.00000	0.00000
1.1	0.09091	0.09531	1.1	0.09529	0.09531	1.1	0.09531	0.09531
1.2	0.18056	0.18232	1.2	0.18229	0.18232	1.2	0.18232	0.18232
1.3	0.25740	0.26236	1.3	0.26233	0.26236	1.3	0.26236	0.26236
1.4	0.32653	0.33647	1.4	0.33635	0.33647	1.4	0.33646	0.33647

Рисунок 4 – результаты на скорректированном диапазоне.

Выводы

В рамках данной контрольной работы были закреплены навыки работы с языком С, я получил возможность продемонстрировать знание и понимание языковых конструкций и базовых алгоритмов.