

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4

Рекурсия
Тема

Преподаватель

подпись, дата

А. С. Черниговский

инициалы, фамилия

Студент КИ19-17/16 031939175

номер группы, зачетной
книжки

подпись, дата

А. Д. Непомнящий

инициалы, фамилия

Красноярск 2020

1 Цель

Цель настоящей работы состоит в том, чтобы продолжить ознакомление с синтаксисом языка СИ, стандартом кодирования, изучить принципы работы функций СИ и применение рекурсии.

2 Задачи

Для выполнения лабораторной работы необходимо выполнить следующие задачи:

На оценку 3 балла:

- 1) написать программу в соответствии с заданием;
- 2) разбить программу на функции;

На оценку 4 балла:

- 1) выполнить требования предыдущих пунктов;
- 2) добавить проверку входных аргументов;
- 3) не допустить использование глобальных переменных;

На оценку 5 баллов:

- 1) выполнить требования предыдущих пунктов;
- 2) добавить повторение программы по желанию пользователя.

3 Описание задания

Дано натуральное число $n > 2$. Необходимо вычислить n -е простое число Ньюмена-Шэнкса-Уильямса, используя рекуррентное соотношение. Сравнить результат с результатом, полученным с помощью прямой формулы.

4 Ход выполнения

Ниже представлен листинг программы по заданию.

Листинг 1 – Код программы, решающей задачу

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <math.h>

enum OperationsCodes
{
    // Перечисление кодов операций для организации главного меню.
    CALCULATE = 1,
    QUIT = 2
};

enum Sizes
{
    // Перечисление наибольших размеров массивов.
    INPUT_SIZE = 100
};

int CycleInputInt(char* stringToOutput, bool(* pChecker)(int))
{
    // Функция для ввода целого числа с проверкой ввода.
    //
    // char* stringToOutput - строка, которую нужно выводить
    // ... в запросе ввода;
    // bool(* pChecker)(int) - указатель на функцию, проверяющую
    // ... дополнительные условия.

    int number;    // Необходимое число
    int position;  // Позиция числа в введенной строке
    char input[INPUT_SIZE];    // Строка для ввода

    // Считывает и проверяет ввод по нескольким условиям, до тех пор,
    // пока не будет введено корректно.
    while (true)
    {
        printf("%s\n", stringToOutput);
        fflush(stdout);
```

Продолжение листинга 1

```
char* fgetsRet = fgets(input, INPUT_SIZE, stdin);
if (fgetsRet == NULL)
{
    printf("Wrong format!\n");
    continue;
}
int inputLength = strlen(input) - 1;
input[inputLength] = '\0';
int sscanfRet = sscanf(input, "%d%n", &number, &position);
if (position != inputLength)
{
    printf("Wrong format!\n");
    continue;
}
if (pChecker && !pChecker(number))
{
    printf("Wrong format!\n");
    continue;
}
if (sscanfRet == 1) break;
printf("Wrong format!\n");
}
return number;
}

bool OperationInputChecker(int operationCode)
{
    // Функция для вызова в функциях ввода с проверкой.
    // ... Возвращает true, если введенное значение может быть
    // значением кода операции в меню.
    //
    // int operationCode - число, которое нужно проверить.

    return operationCode >= CALCULATE && operationCode <= QUIT;
}

bool NSWPrimeArgChecker(int numberToCheck)
{
    // Функция для вызова в функциях ввода с проверкой.
    // ... Возвращает true, если введенное значение может быть
    // значением N для задачи.
```

Продолжение листинга 1

```
//
// int numberToCheck - число, которое нужно проверить.
return numberToCheck > 2;
}

int RecNSWPrime(int n)
{
    // Функция вычисляет n-ое NSW-число через рекуррентное соотношение
    //
    // int n - порядок нужного члена последовательности NSW-чисел.
    if (n < 2)
    {
        return 1;
    }
    return 2 * RecNSWPrime(n - 1) + RecNSWPrime(n - 2);
}

float NSWPrime(int n)
{
    // Функция вычисляет n-ое NSW-число по явной формуле
    //
    // int n - порядок нужного члена последовательности NSW-чисел.
    return (powf(1 + sqrt(2), n) + powf(1 - sqrt(2), n)) / 2;
}

int main()
{
    int operationCode;
    while (true)
    {
        printf("\n1. Calculate with new N.\n"
               "2. Quit.\n");
        operationCode = CycleInputInt(
            "Choose the command and enter its number",
            OperationInputChecker);

        // Получение числа N и проведение всех вычислений
        if (operationCode == CALCULATE)
        {
            int arg;
            float res;
```

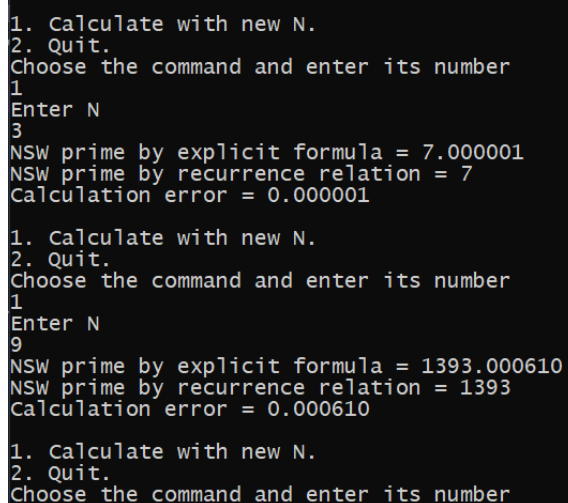
Окончание листинга 1

```
        int recRes;
        arg = CycleInputInt("Enter N", NSWPrimeArgChecker);
        res = NSWPrime(arg);
        recRes = RecNSWPrime(arg);
        printf("NSW prime by explicit formula = %f\n", res);
        printf("NSW prime by recurrence relation = %d\n", recRes);
        printf("Calculation error = %f\n", fabsf(res - recRes));
    }

    // Выход из меню
    if (operationCode == QUIT)
    {
        break;
    }
}
return 0;
}
```

5 Результат

Ниже представлены скриншоты с консольным выводом.



```
1. Calculate with new N.  
2. Quit.  
Choose the command and enter its number  
1  
Enter N  
3  
NSW prime by explicit formula = 7.000001  
NSW prime by recurrence relation = 7  
Calculation error = 0.000001  
  
1. Calculate with new N.  
2. Quit.  
Choose the command and enter its number  
1  
Enter N  
9  
NSW prime by explicit formula = 1393.000610  
NSW prime by recurrence relation = 1393  
Calculation error = 0.000610  
  
1. Calculate with new N.  
2. Quit.  
Choose the command and enter its number
```

Рисунок 1 – Проведение вычислений с двумя разными N

6 Выводы

Была написана программа, соответствующая поставленным задачам. В ходе работы были изучены некоторые аспекты синтаксиса языка программирования Си, в частности – принципы работы с функциями. Была осознана такая концепция, как рекурсия. Код был отформатирован в соответствии со стандартом, принятым в учебном заведении.