

ГУАП

КАФЕДРА № 53

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____
ПРЕПОДАВАТЕЛЬ

старший преподаватель
должность, уч. степень, звание

подпись, дата

Ушаков В.А.
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Рекурсивный алгоритм

Вариант 11

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ(А)

СТУДЕНТ ГР. № 5138

подпись, дата

Воробьев В.А.
инициалы, фамилия

Санкт-Петербург 2022

Задание: реализовать программную функцию на языке C/C++, выполняющую поставленную задачу. Написать код функции, принимающей в качестве аргументов и возвращающей все необходимые параметры, без использования глобальных переменных. Допустимо использование дополнительных функций. Из наименования функции и принимаемых аргументов должно быть ясно их назначение. В ходе тестирования функции при каждом вызове рекурсивной функции необходимо вывести отладочную информацию: порядковый номер вызова рекурсивной функции, значения изменяющегося аргумента и возвращаемого значения, если они присутствуют. Привести глубину рекурсии для каждого тестового примера.

11	Реализовать рекурсивную функцию вычисления $S(x, N) = \begin{cases} S(x, N-1) + 2, & \text{при четных } N \\ S(x, N-1) * 2, & \text{при нечетных } N \end{cases}$ $S(x, 0) = x$ по заданным N, x .	6, 2	30
----	---	------	----

Рисунок 1 - вариант задания

Выполнение задания:

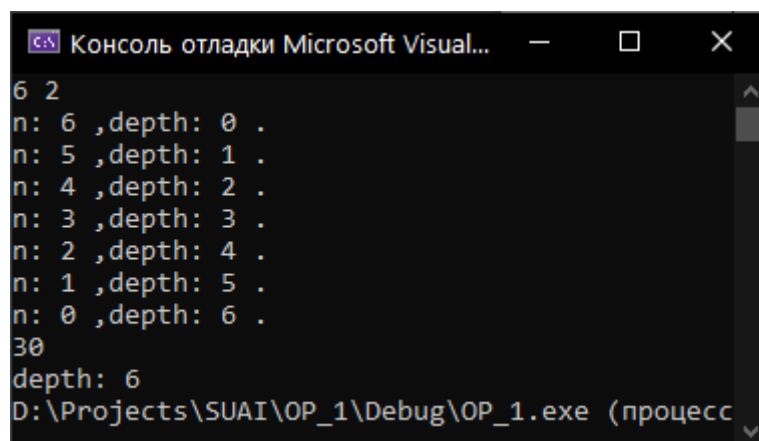
```
#include <iostream>
#include <vector>
#include <string>
#include <math.h>

using namespace std;

// Не понимаю, зачем тут считать глубину.
int S(int x, int n, int* memory, int depth = 0) {
    if (*memory < depth) *memory = depth;
    cout << "n: " << n << " ,depth: " << depth << " .\n";
    if (n == 0) {
        return x;
    }
    else if (n % 2 == 0) {
        return S(x, n - 1, memory, depth + 1) + 2;
    }
    else {
        return S(x, n - 1, memory, depth + 1) * 2;
    }
}

void main()
{
    int x, n;
    cin >> n >> x;
    int depth;
    int res = S(x, n, &depth);
    cout << res << "\ndepth: " << depth;
}
```

Рисунок 2 – код



```
Microsoft Visual...
6 2
n: 6 ,depth: 0 .
n: 5 ,depth: 1 .
n: 4 ,depth: 2 .
n: 3 ,depth: 3 .
n: 2 ,depth: 4 .
n: 1 ,depth: 5 .
n: 0 ,depth: 6 .
30
depth: 6
D:\Projects\SUAI\OP_1\Debug\OP_1.exe (процесс
```

Рисунок 3 - ввод и вывод

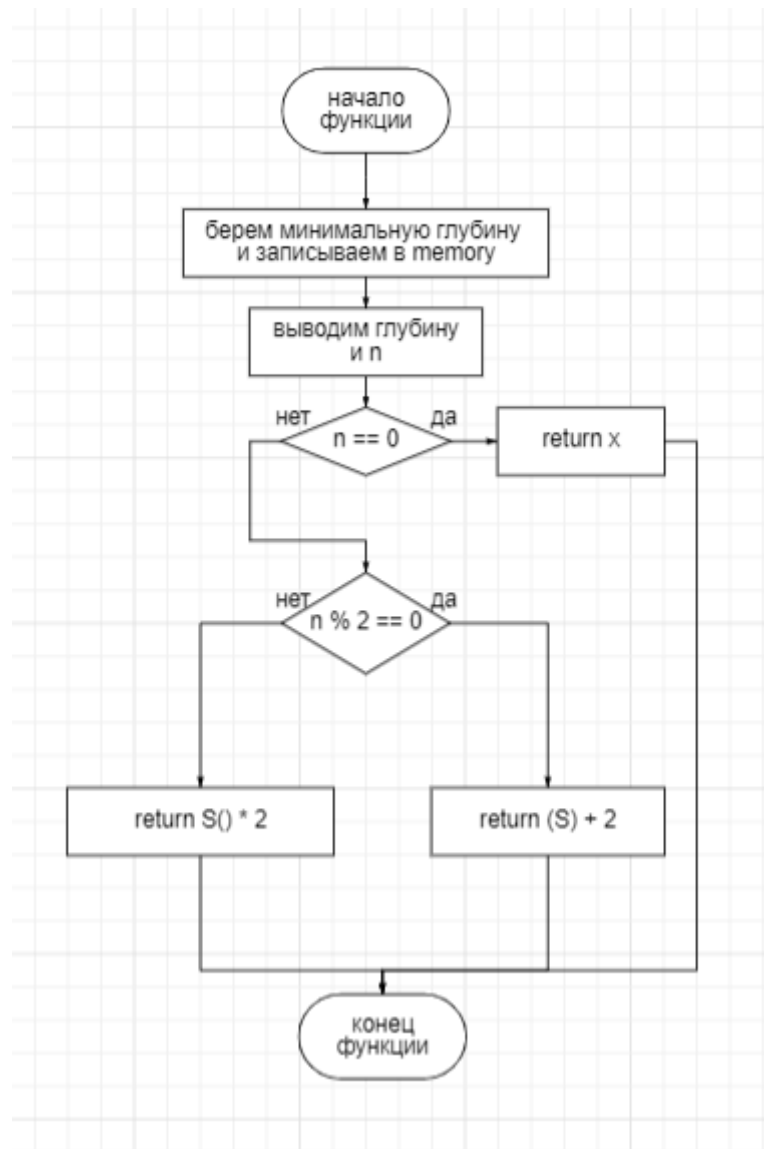


Рисунок 4 - Блок-схема рекурсивной функции



Рисунок 5 - Блок-схема основной программы

Вывод: в ходе выполнения работы научился создавать рекурсивные алгоритмы. Понял понятие “глубина рекурсии”, а также её вычисление в самой рекурсии.