Сравнение циклов и рекурсии

Цели

- Оценить недостатки процедурного программирования
- Научиться строить рекурсивные алгоритмы

Порядок выполнения

1. Написать программу по заданию с использованием цикла
2. Провести трассировку программы
3. Составить рекурсивную функцию для решения выданного задания
4. Реализовать составленную рекурсивную функцию на языке программирования
5. Написать отчет

Рекомендации по выполнению

- Массивы фиксированной длины
- Трассировка отключается макросом
- Данные задаются внутри исходного кода

Состав отчета

- Титульный лист (фамилия, группа, номер варианта, наименование работы, задание)
- Текст рекурсивной функции
- Текст итеративной функции
- Результаты выполнения

Варианты заданий

1. Напишите программу печатающую n-ое число Фибоначчи.
2. Напишите программу вычисляющую факториал натурального числа.
3. Напишите программу перемножающую два целых неотрицательных числа без использования операции умножения.
4. Напишите программу, печатающую значение многочлена степени $n \geq 0$ в заданной точке x_0. Коэффициенты многочлена хранятся в массиве a в порядке убывания степений и являются целыми числами, так же как и значение x_0.
5. Напишите программу печатающую значение производной многочлена степени $n \geq 0$ в заданной точке x_0. Коэффициенты многочлена хранятся в массиве a в порядке убывания степений и являются целыми числами, так же как и значение x_0.
6. Напишите программу возводящую целое число в целую неотрицательную степень.
7. Напишите программу принимающую на вход натуральное число и выводающую Yes если число является простым, и No - если не является.
8. Напишите программу генерации всех правильных скобочных структур длины $2n$. Например для $n = 3$ таких структур может быть 5: ()()(), (())(), ()(()), ((())), (()()).
9. Имеется три стержня A, B, C. На стержень A нанизано n дисков радиуса $1, 2, ..., n$ таким образом, что диск радиуса i является i-м сверху. Требуется переместить все диски на стержень B, сохраняя их порядок расположения (диск с большим радиусом находится ниже). За один раз можно перемещать только один диск с любого стержня на любой другой стержень. При этом должно выполняться следующее условие: на каждом стержне ни в какой момент времени никакой диск не может находиться выше диска с меньшим радиусом.
10. Напишите программу выводящую сумму квадратов всех натуральных чисел от 1 до введённого n.
11. Напишите программу печатающую n-ое простое число.

12. Напишите программу, печатающую старшую цифру в десятичной записи введенного натурального числа.

13. Напишите программу, печатающую количество цифр в десятичной записи введенного натурального числа.

14. Напишите программу, печатающую количество натуральных решений неравенства $x^2 + y^2 < n$ для введенного n.

15. Напишите программу, вводящую натуральное число , и печатающую количество точек с целочисленными координатами внутри замкнутого шара радиуса с центром в начале координат.

16. Напишите программу, печатающую квадраты всех целых чисел от нуля до введенного натурального n, не использующую операций умножения.

17. Напишите программу, находящую количество счастливых билетов с шестизначными номерами. Билет называется счастливым, если сумма его первых трех цифр равна сумме трех последних.

Пример

Задание Напишите программу проверяющую является ли введённое число факториалом какого либо числа.

Итеративное решение Возьмём математическое определение факториала:

$$n! = 1 \cdot 2 \cdot \ldots \cdot n = \prod_{i=1}^{n} i \tag{1}$$

Получается что факториал числа n должен делиться нацело на все натуральные числа до n включая n. Напишем программу рализующую такую проверку:

```
#include <stdio.h>

int check_factorial_iterate(int number){
    if(number < 0)
        return 0;
    if(number == 0)
        return 1;
    int i = 1;
    int n = 1;
    for(;n<number;n*=i,i++){
```

```
            if(number%i != 0)
                return 0;
        }
        return 1;
}

int main(){
        if(check_factorial_iterate(362880))
            printf("%s","yes");
        else
            printf("%s","no");
        return 0;
}
```

Listing 1: итеративная программа

Рекурсивное решение Возьмём рекурсивное определение факториала:

$$n! = \begin{cases} 1 & n = 0, \\ n \cdot (n-1)! & n > 0. \end{cases} \tag{2}$$

```
#include <stdio.h>

int check_factorial_recursive(int number, int i){
        if(number < 0)
            return 0;
        if(number == 0)
            return 1;

        if(number == 1)
            return 1;

        if( number%i != 0)
            return 0;
        else
            return check_factorial_recursive(number/i,i+1);

}

int main(){
        int number = 362881;

        if(check_factorial_recursive(number,1))
            printf("%s\n","yes");
        else
            printf("%s\n","no");
```

```
    return 0;
}
```

Listing 2: рекурсивная программа