

## **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**

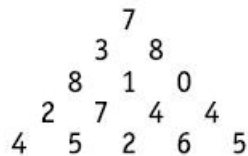
- Массивы фиксированной длины
- Трассировка отключается макросом
- Данные задаются внутри исходного кода

### **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 проверяющую является ли введённое число простым.
4. Напишите программу сортировки массива.
5. Напишите программу поиска максимального элемента массива.
6. Напишите программу возводящую одно натуральное число в степень другого, без использования функции возведения в степень.
7. Напишите программу генерации всех правильных скобочных структур длины  $2n$ . Например для  $n = 3$  таких структур может быть 5:  $()()()$ ,  $((()))$ ,  $()(())$ ,  $((()))$ ,  $((()))$ .
8. Имеется три стержня А, В, С. На стержень А нанизано  $n$  дисков радиуса  $1, 2, \dots, n$  таким образом, что диск радиуса  $i$  является  $i$ -м сверху. Требуется переместить все диски на стержень В, 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.
9. Заданы три числа:  $a$ ,  $b$ ,  $c$ . Необходимо выяснить, можно ли так переставить цифры в числах  $a$  и  $b$ , чтобы в сумме получилось  $c$ .
10. Задан набор слов. Построить из них любую цепочку таким образом, чтобы символ в конце слова совпадал с символом в начале следующего.
11. На рисунке показан пример треугольника из чисел. Написать программу, вычисляющую наибольшую сумму чисел, через которые проходит путь, начинающийся на вершине и заканчивающийся где-то на основании.



## Пример

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

**Итеративное решение** Возьмём математическое определение факториала:

$$n! = 1 \cdot 2 \cdot \dots \cdot n = \prod_{i=1}^n i \quad (1)$$

Получается что факториал числа  $n$  должен делиться нацело на все натуральные числа до  $n$  включая  $n$ . Напишем программу реализующую 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:

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

---

```
#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