

Алгоритмы

Задачи для урока

1. «Восхождение на Факториальную гору»

Мудрец установил на вершине башню: чтоб попасть внутрь, нужно вычислить $n!$.

Напишите рекурсивную функцию `long factorial(int n)`.

Вызов: `factorial(5) → 120`

2. «Числовой дракон»

Дракон хранит на своей латы цифру « n -е число Фибоначчи».

Напишите рекурсивную функцию `int fib(int n)`.

Вызов: `fib(7) → 13`

3. «Жезл возведения в степень»

Магический посох возводит число A в степень B через волшебный ритуал:

```
int power(int a, int b) {  
    if (b == 0) return 1;  
    return a * power(a, b - 1);  
}
```

Допишите эту функцию и протестируйте для разных A , B .

4. «Счетчик вселенной»

Ученые в симуляции хотят узнать сумму чисел от 1 до N , но без рекурсии.

Напишите функцию `long sumIter(int n)`, которая с помощью цикла `for` подсчитывает сумму.

5. «Пчелиный рой Фибоначчи»

Пчёлы строят ульи по ряду Фибоначчи. Нужно вывести первые N чисел последовательности через пробел, используя только цикл `for` (без `row`, без рекурсии).

6. «Лабиринт цифр»

В таинственном лабиринте каждый шаг — отделение последней цифры числа.

Напишите рекурсивную функцию `void printDigits(int n)`, которая выводит цифры числа `n` в обратном порядке (с пробелом), вызывая себя для оставшейся части:

```
void printDigits(int n) {  
  
    printf("%d ", n % 10);  
  
    if (n / 10 > 0) printDigits(n / 10);  
  
}
```

Домашнее задание

1. «Факториал 2.0»

Повторите задачу 1, но реализуйте `factorialIter(int n)` через цикл `for`.

2. «Улей Фибоначчи»

Повторите вывод Фибоначчи (5 урока), но напишите `fibIter(int n)` с одним циклом `for`.

3. «Сумма цифр ритуалом»

Напишите рекурсивную функцию `int sumDigitsRec(int n)`, которая возвращает сумму цифр числа, пока не дойдёт до 0.

4. «Мощь степеней»

Реализуйте `int powerIter(int a, int b)`, которая возводит `a` в степень `b` через цикл `for`.

5. «Охотник за простыми»

Напишите `int isPrime(int n)`, которая проверяет, простое ли число `n`, перебором делителей от 2 до `n-1`, и выводит 1 (да) или 0 (нет).