

# Технология программирования на ЭВМ

## Функции, рекурсия

Баев А.Ж.

Казахстанский филиал МГУ

03 декабря 2019

## Пример: факториал

Дано целое  $n$  от 1 до 10. Вычислить  $n!$ .

Выпишем рекуррентное соотношение

$$n! = n * (n - 1)!$$

Выпишем начальное значение

$$1! = 1$$

Аналогия с методом математической индукции!

## Пример: факториал

```
1  int factorial(int n) {
2      if (n == 1) {
3          return 1;
4      }
5      int answer = factorial(n-1) * n;
6      return answer;
7  }
8
9  int main() {
10     int result = factorial(6);
11     return 0;
12 }
```

## Пример: факториал

```
1 int factorial(int n) {  
2     int answer = factorial(n-1) * n;  
3     return answer;  
4 }
```

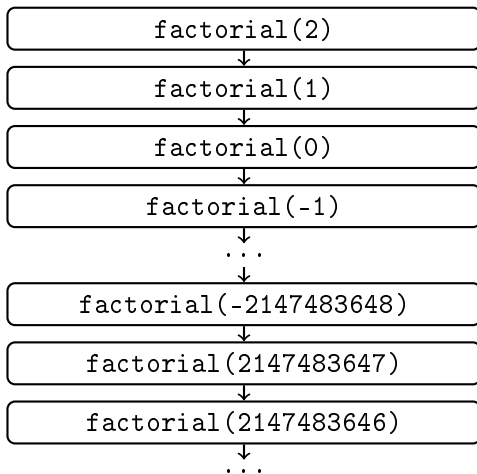
Какая будет ошибка?

```
1 $ segmentation fault
```

Почему?

## Пример: факториал

Напишем порядок запуска:



Заканчивается системный стек, где хранятся аргументы.

## Пример: вывод чисел в прямом порядке

Дана строка, вывести её же.

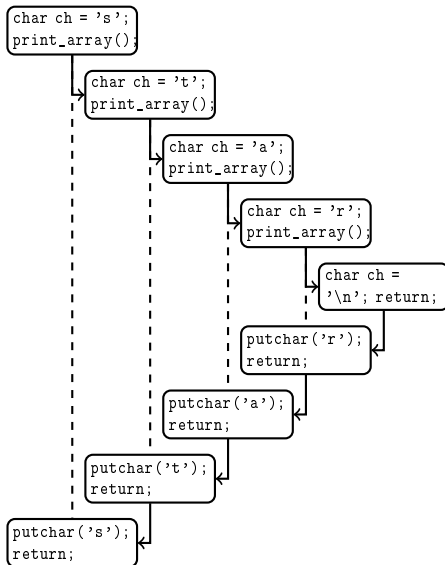
```
1 void print_array() {  
2     char ch = getchar();  
3     if (ch == '\n') {  
4         return;  
5     }  
6     putchar(ch);  
7     print_array();  
8 }
```

## Пример: вывод чисел в прямом порядке

Дана строка, вывести её в обратном порядке.

```
1 void print_array() {  
2     char ch = getchar();  
3     if (ch == '\\n') {  
4         return;  
5     }  
6     print_array();  
7     putchar(ch);  
8 }
```

## Пример: вывод чисел в прямом порядке





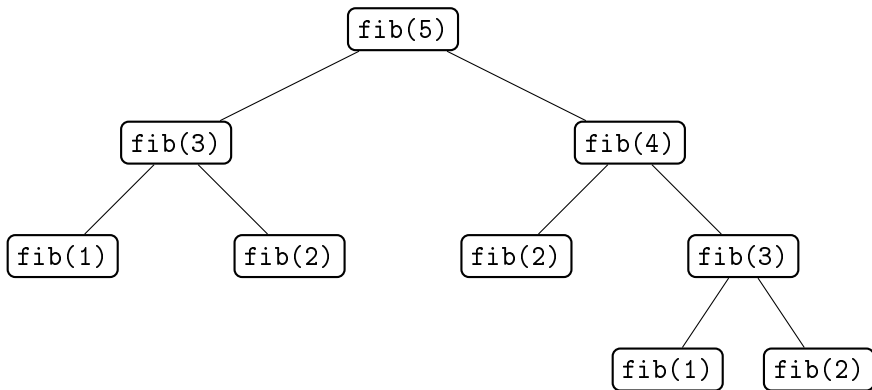
## Пример: фибоначчи

Дано  $n$  от 1 до 90. Вывести  $F_n$ , где  $F_n = F_{n-1} + F_{n-2}$ ,  
 $F_1 = F_2 = 1$ .

```
1 long long fib(int n) {
2     if (n <= 2) {
3         return 1;
4     }
5     long long answer = fib(n-1) + fib(n-2);
6     return answer;
7 }
8
9 int main() {
10     int n;
11     scanf("%d", &n);
12     printf("%lld", fib(n));
13     return 0;
14 }
```

## Пример: фибоначчи

Вычисления будут выполнены следующим образом:



Что плохо?

## Пример: НОД

Даны два целых числа  $a$  и  $b$  от 0 до  $10^9$ , причем хотя бы одно из них отлично от нуля. Написать рекурсивную функцию, которая вычисляет наибольший общий делитель.

Ввод:

800 1024

Вывод:

32

## Пример: НОД

Алгоритм Евклида (при  $b \neq 0$ ):

$$(a; b) = (b; a \bmod b)$$

Например:

$$\begin{aligned}(800; 1024) &= (1024; 800 \bmod 1024) = \\&= (1024; 800) = (800; 1024 \bmod 800) = \\&= (800; 224) = (224; 800 \bmod 224) = \\&= (224; 128) = (128; 224 \bmod 128) = \\&= (128; 96) = (96; 128 \bmod 96) = \\&= (96; 32) = (32; 96 \bmod 32) = (32; 0) = 32\end{aligned}$$

## Пример: НОД

```
1  #include <stdio.h>
2  int gcd(int a, int b) {
3      if (b == 0) {
4          return a;
5      }
6      int d = gcd(b, a % b);
7      return d;
8  }
9  int main() {
10     int a, b;
11     scanf("%d %d", &a, &b);
12     printf("%d\n", gcd(a, b));
13     return 0;
14 }
```

## Пример: степень

Даны два целых числа  $a$  от 0 до  $10^9$  и  $n$  от 0 до  $10^4$ , причем хотя бы одно из них отлично от нуля. Написать рекурсивную функцию, которая вычисляет последние 2 цифры числа  $a^n$ .

Ввод:

3 11

Вывод:

47

## Пример: степень

$$a^n = \begin{cases} a^{[n/2]} \cdot a^{[n/2]} \cdot a, & n \text{ — нечетное} \\ a^{[n/2]} \cdot a^{[n/2]}, & n \text{ — четное, больше 0} \\ 1, & n = 0 \end{cases}$$

Например

$$a^{25} = a^{12} \cdot a^{12} \cdot a$$

$$a^{12} = a^6 \cdot a^6$$

$$a^6 = a^3 \cdot a^3$$

$$a^3 = a^1 \cdot a^1 \cdot a$$

$$a^1 = a^0 \cdot a^0 \cdot a$$

## Пример: степень

```
1  int last_digits(int a, int k) {  
2      if (k == 0) {  
3          return 1;  
4      }  
5      int answer = last_digits(a, k / 2);  
6      answer = answer * answer % 100;  
7      if (k % 2 == 1) {  
8          answer = answer * a % 100;  
9      }  
10     return answer;  
11 }
```