Технология программирования на ЭВМ
Функции, рекурсия

Баев А.Ж.

Казахстанский филиал МГУ

29 ноября 2018

Пример: факториал

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

Выпишем рекуррентое соотношение

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

Выпишем начальное значение

$$1! = 1$$

Аналогия с методом математической индукции!

Пример: факториал

```
int factorial(int n) {
    if (n == 1) {
        return 1;
    }
    int answer = factorial(n-1) * n;
    return answer;
}

int main() {
   int result = factorial(6);
   return 0;
}
```

Пример: факториал

```c
int factorial(int n) {
    int answer = factorial(n-1) * n;
    return answer;
}
```

Какая будет ошибка?

```
$ segmentation fault
```

Почему?

Пример: факториал

Напишем порядок запуска:

```
factorial(2)
     ↓
factorial(1)
     ↓
factorial(0)
     ↓
factorial(-1)
     ↓
    ...
     ↓
factorial(-2147483648)
     ↓
factorial(2147483647)
     ↓
factorial(2147483646)
     ↓
    ...
```

Заканчивается системный стек, где хранятся аргументы.

Пример: вывод чисел в прямом порядке

Дана строка, вывести её же.

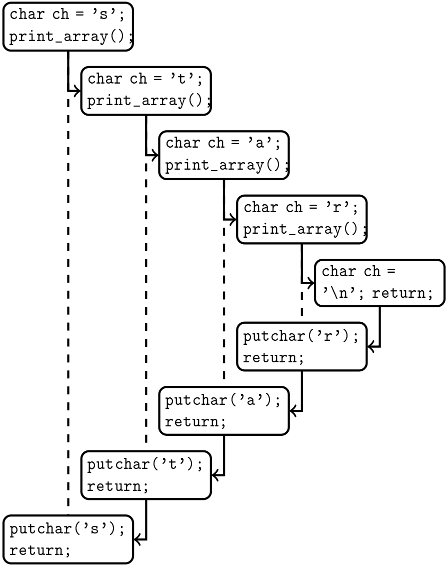
```c
void print_array() {
    char ch = getchar();
    if (ch == '\n') {
        return;
    }
    putchar(ch);
    print_array();
}
```

Пример: вывод чисел в прямом порядке

Дана строка, вывести её в обратном порядке.

```c
void print_array() {
    char ch = getchar();
    if (ch == '\n') {
        return;
    }
    print_array();
    putchar(ch);
}
```

Пример: вывод чисел в прямом порядке



Пример: фибоначчи

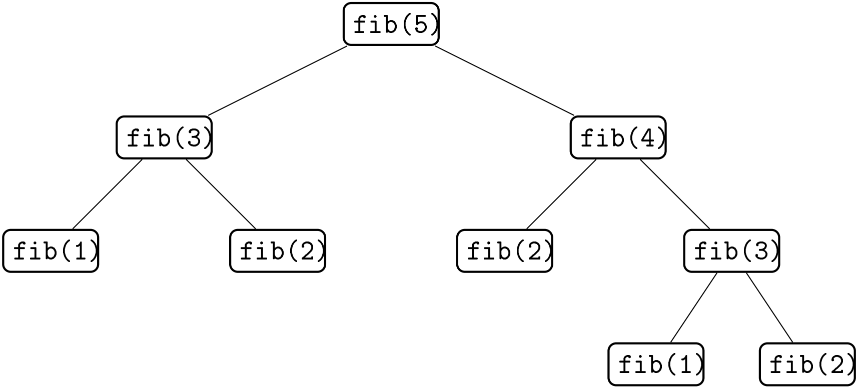
Дано n от 1 до 90. Вывести F_n, где $F_n = F_{n-1} + F_{n-2}$, $F_1 = F_2 = 1$.

```
long long fib(int n) {
    if (n <= 2) {
        return 1;
    }
    long long answer = fib(n-1) + fib(n-2);
    return answer;
}

int main() {
    int n;
    scanf("%d", &n);
    printf("%lld", fib(n));
    return 0;
}
```

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:

$$(800; 1024) = (1024; 800 \bmod 1024) =$$

$$= (1024; 800) = (800; 1024 \bmod 800) =$$

$$= (800; 224) = (224; 800 \bmod 224) =$$

$$= (224; 128) = (128; 224 \bmod 128) =$$

$$= (128; 96) = (96; 128 \bmod 96) =$$

$$= (96; 32) = (32; 96 \bmod 32) = (32; 0) = 32$$

Пример: НОД

```c
#include <stdio.h>
int gcd(int a, int b) {
    if (b == 0) {
        return a;
    }
    int d = gcd(b, a % b);
    return d;
}
int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", gcd(a, b));
    return 0;
}
```

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

```
int last_digits(int a, int k) {
    if (k == 0) {
        return 1;
    }
    int answer = last_digits(a, k / 2);
    answer = answer * answer % 100;
    if (k % 2 == 1) {
        answer = answer * a % 100;
    }
    return answer;
}
```