

Задачи 3.3. Рекурсивные функции

- 1) Описать рекурсивные функции Fact и Fact2, вычисляющие значения факториала $N!$ и двойного факториала $N!!$ соответственно ($N > 0$ — параметр целого типа).
- 2) Описать рекурсивную функцию PowerN, находящую значение n -й степени числа x по формуле:

$$x^0 = 1, x^n = x \cdot x^{n-1} \text{ при } n > 0, \quad x^n = \frac{1}{x^{-n}} \text{ при } n < 0$$

($x \geq 0$ — вещественное число, n — целое).

- 3) Описать рекурсивную функцию SqrtK(x, k, n), находящую приближенное значение корня k -й степени из числа x по формуле:

$$y(0) = 1, \quad y(n+1) = y(n) - (y(n) - \frac{x}{y^{k-1}(n)})/k,$$

где $y(n)$ обозначает SqrtK(x, k, n) (x — вещественный параметр, k и n — целые; $x > 0, k > 1, n > 0$).

- 4) Описать рекурсивную функцию FibRes, вычисляющую N -е число Фибоначчи $F(N)$ по формуле:

$$F(1) = F(2) = 1, F(k) = F(k-2) + F(k-1), k = 3, 4, \dots$$

С помощью этой функции найти пять чисел Фибоначчи с указанными номерами и вывести эти числа вместе с количеством рекурсивных вызовов функции FibRes, потребовавшихся для их нахождения.

- 5) Описать рекурсивную функцию C(m, n), находящую число сочетаний из n элементов по m , используя формулу:

$$C(0, n) = C(n, n) = 1, C(m, n) = C(m, n-1) + C(m-1, n-1)$$

при $0 < m < n$ (m и n — целые параметры; $n > 0, 0 \leq m \leq n$).

Дано число N и пять различных значений M . Вывести числа $C(M, N)$ вместе с количеством рекурсивных вызовов функции C , потребовавшихся для их нахождения.

- 6) Описать рекурсивную функцию NOD(A, B), находящую наибольший общий делитель двух натуральных чисел A и B , используя алгоритм Евклида:

$$\text{NOD}(A, B) = \text{NOD}(B \bmod A, A), \text{ если } A \neq 0,$$

$$\text{NOD}(0, B) = B.$$

- 7) С помощью этой функции найти наибольшие общие делители пар A и B , A и C , A и D , если даны числа A, B, C, D .
- 8) Описать рекурсивную функцию MinRes, которая находит минимальный элемент массива A размера N , не используя оператор цикла. С помощью этой функции найти минимальные элементы трёх заданных массивов.
- 9) Описать рекурсивную функцию Digits, находящую количество цифр в строке S без использования оператора цикла. С помощью этой функции найти количество цифр в заданной строке.

- 10) Описать рекурсивную функцию *Simm*, проверяющую, является ли симметричной строка S , без использования оператора цикла. С помощью этой функции проверить заданную строку.
- 11) Задано положительное и отрицательное число в двоичной системе. Составить программу вычисления суммы этих чисел, используя функцию сложения чисел в двоичной системе счисления.
- 12) Описать рекурсивную функцию *Root*, которая методом деления отрезка пополам находит с точностью ε корень уравнения $f(x) = 0$ на отрезке $[a, b]$ (считать, что $\varepsilon > 0$, $a < b$, $f(a) - f(b) < 0$ и $f(x)$ — непрерывная и монотонная на отрезке $[a, b]$ функция).
- 13) Дано число n , десятичная запись которого не содержит нулей. Получите число, записанное теми же цифрами, но в противоположном порядке. При решении этой задачи нельзя использовать циклы, строки, списки разрешается только рекурсия и целочисленная арифметика. Функция должна возвращать целое число, являющееся результатом работы программы, выводить число по одной цифре нельзя.
- 14) Описать рекурсивную логическую функцию *Simm*, проверяющую, является ли симметричной часть строки S , начинающаяся i -м и заканчивающаяся j -м её элементами.
- 15) Дано натуральное число N . Вычислите сумму его цифр. При решении этой задачи нельзя использовать строки, списки и циклы.
- 16) Дано слово, состоящее только из строчных латинских букв. Проверьте, является ли это слово палиндромом. Выведите *yes* или *no*. При решении этой задачи нельзя пользоваться циклами, также нельзя использовать срезы с шагом, отличным от 1.
- 17) Описать рекурсивную функцию *BinarySearch*, которая находит индекс заданного элемента в отсортированном массиве.
- 18) Описать рекурсивную функцию *GCD*, находящую наибольший общий делитель двух чисел A и B с помощью алгоритма Евклида.
- 19) Описать рекурсивную функцию *Palindrome*, которая проверяет, является ли заданная строка палиндромом.
- 20) Описать рекурсивную функцию *SumDigits*, вычисляющую сумму цифр числа.
- 21) Описать рекурсивную функцию *BinaryToDecimal*, которая переводит число из двоичной системы счисления в десятичную.
- 22) Описать рекурсивную функцию *CountVowels*, находящую количество гласных букв в строке S без использования оператора цикла. С помощью этой функции найти количество гласных букв в заданной строке.

- 23) Описать рекурсивную функцию CountConsonants, находящую количество согласных букв в строке S без использования оператора цикла. С помощью этой функции найти количество согласных букв в заданной строке.
- 24) Описать рекурсивную функцию CountSpaces, находящую количество пробелов в строке S без использования оператора цикла. С помощью этой функции найти количество пробелов в заданной строке.
- 25) Описать рекурсивную функцию CountUppercase, находящую количество прописных букв в строке S без использования оператора цикла. С помощью этой функции найти количество прописных букв в заданной строке.
- 26) Описать рекурсивную функцию CountLowercase, находящую количество строчных букв в строке S без использования оператора цикла. С помощью этой функции найти количество строчных букв в заданной строке.
- 27) Описать рекурсивную функцию CountDigits, находящую количество цифр в строке S без использования оператора цикла. С помощью этой функции найти количество цифр в заданной строке.
- 28) Описать рекурсивную функцию CountSymbols, находящую количество определенного символа в строке S без использования оператора цикла. С помощью этой функции найти количество определенного символа в заданной строке.
- 29) Описать рекурсивную функцию IsSubstringInOrder, которая проверяет, является ли одна строка подстрокой другой строки в определенной последовательности символов.
- 30) Описать рекурсивную функцию IsAlternating, которая проверяет, является ли заданная строка последовательностями символов, чередующихся по типу (буква-цифра-буква-цифра и т.д.).
- 31) Описать рекурсивную функцию CountOccurrence, находящую количество вхождений подстроки в строке S без использования оператора цикла. С помощью этой функции найти количество вхождений подстроки в заданной строке.