

формулам оказываются весьма неэффективными (см., например, задания Recur4 и Recur6). Однако именно на подобных примерах проще всего получить первоначальные навыки разработки рекурсивных алгоритмов.

**Recur1°.** Описать рекурсивную функцию  $\text{Fact}(N)$  вещественного типа, вычисляющую значение *факториала*

$$N! = 1 \cdot 2 \cdot \dots \cdot N$$

( $N > 0$  — параметр целого типа). С помощью этой функции вычислить факториалы пяти данных чисел.

**Recur2.** Описать рекурсивную функцию  $\text{Fact2}(N)$  вещественного типа, вычисляющую значение *двойного факториала*

$$N!! = N \cdot (N-2) \cdot (N-4) \cdot \dots$$

( $N > 0$  — параметр целого типа; последний сомножитель в произведении равен 2, если  $N$  — четное число, и 1, если  $N$  — нечетное). С помощью этой функции вычислить двойные факториалы пяти данных чисел.

**Recur3.** Описать рекурсивную функцию  $\text{PowerN}(X, N)$  вещественного типа, находящую значение  $N$ -й степени числа  $X$  по формулам:

$$\begin{aligned} X^0 &= 1, \\ X^N &= (X^{N/2})^2 \text{ при четных } N > 0, \\ X^N &= X \cdot X^{N-1} \text{ при нечетных } N > 0, \\ X^N &= 1/X^{-N} \text{ при } N < 0 \end{aligned}$$

( $X \neq 0$  — вещественное число,  $N$  — целое; в формуле для четных  $N$  должна использоваться операция *целочисленного деления*). С помощью этой функции найти значения  $X^N$  для данного  $X$  при пяти данных значениях  $N$ .

**Recur4.** Описать рекурсивную функцию  $\text{Fib1}(N)$  целого типа, вычисляющую  $N$ -й элемент последовательности *чисел Фибоначчи* ( $N$  — целое число):

$$F_1 = F_2 = 1, \quad F_K = F_{K-2} + F_{K-1}, \quad K = 3, 4, \dots$$

С помощью этой функции найти пять чисел Фибоначчи с данными номерами, и вывести эти числа вместе с количеством рекурсивных вызовов функции  $\text{Fib1}$ , потребовавшихся для их нахождения.

**Recur5.** Описать рекурсивную функцию  $\text{Fib2}(N)$  целого типа, вычисляющую  $N$ -й элемент последовательности *чисел Фибоначчи* ( $N$  — целое число):

$$F_1 = F_2 = 1, \quad F_K = F_{K-2} + F_{K-1}, \quad K = 3, 4, \dots$$

Считать, что номер  $N$  не превосходит 20. Для уменьшения количества рекурсивных вызовов по сравнению с функцией  $\text{Fib1}$  (см. задание Recur4) создать вспомогательный массив для хранения *уже вычисленных* чисел Фибоначчи и обращаться к нему при выполнении функции  $\text{Fib2}$ . С помощью функции  $\text{Fib2}$  найти пять чисел Фибоначчи с данными номерами.

**Recur6.** Описать рекурсивную функцию  $\text{Combin1}(N, K)$  целого типа, находящую  $C(N, K)$  — *число сочетаний* из  $N$  элементов по  $K$  — с помощью рекуррентного соотношения:

$$C(N, 0) = C(N, N) = 1,$$

$$C(N, K) = C(N-1, K) + C(N-1, K-1) \quad \text{при } 0 < K < N.$$

Параметры функции — целые числа;  $N > 0$ ,  $0 \leq K \leq N$ . Дано число  $N$  и пять различных значений  $K$ . Вывести числа  $C(N, K)$  вместе с количеством рекурсивных вызовов функции `Combin1`, потребовавшихся для их нахождения.

**Recur7.** Описать рекурсивную функцию `Combin2(N, K)` целого типа, находящую  $C(N, K)$  — *число сочетаний* из  $N$  элементов по  $K$  — с помощью рекуррентного соотношения:

$$C(N, 0) = C(N, N) = 1,$$

$$C(N, K) = C(N-1, K) + C(N-1, K-1) \quad \text{при } 0 < K < N.$$

Параметры функции — целые числа;  $N > 0$ ,  $0 \leq K \leq N$ . Считать, что параметр  $N$  не превосходит 20. Для уменьшения количества рекурсивных вызовов по сравнению с функцией `Combin1` (см. задание **Recur6**) описать вспомогательный двумерный массив для хранения *уже вычисленных* чисел  $C(N, K)$  и обращаться к нему при выполнении функции `Combin2`. С помощью функции `Combin2` найти числа  $C(N, K)$  для данного значения  $N$  и пяти различных значений  $K$ .

**Recur8.** Описать рекурсивную функцию `RootK(X, K, N)` вещественного типа, находящую приближенное значение корня  $K$ -й степени из числа  $X$  по формуле:

$$Y_0 = 1, \quad Y_{N+1} = Y_N - (Y_N - X/(Y_N)^{K-1})/K,$$

где  $Y_N$  обозначает `RootK(X, K, N)` при фиксированных  $X$  и  $K$ . Параметры функции:  $X (> 0)$  — вещественное число,  $K (> 1)$  и  $N (> 0)$  — целые. С помощью функции `RootK` найти для данного числа  $X$  приближенные значения его корня  $K$ -й степени при шести данных значениях  $N$ .

**Recur9.** Описать рекурсивную функцию `NOD(A, B)` целого типа, находящую *наибольший общий делитель* (НОД) двух натуральных чисел  $A$  и  $B$ , используя *алгоритм Евклида*:

$$\text{НОД}(A, B) = \text{НОД}(B, A \bmod B), \quad \text{если } B \neq 0; \quad \text{НОД}(A, 0) = A.$$

С помощью этой функции найти `НОД(A, B)`, `НОД(A, C)`, `НОД(A, D)`, если даны числа  $A, B, C, D$ .

**Recur10.** Описать рекурсивную функцию `DigitSum(K)` целого типа, которая находит сумму цифр целого числа  $K$ , не используя оператор цикла. С помощью этой функции найти суммы цифр для пяти данных целых чисел.

**Recur11.** Описать рекурсивную функцию `MaxInt(A, N)` целого типа, которая находит максимальный элемент целочисленного массива  $A$  размера  $N$  ( $1 \leq N \leq 10$ ), не используя оператор цикла. С помощью этой функции найти максимальные элементы массивов  $A, B, C$  размера  $N_A, N_B, N_C$  соответственно.

**Recur12.** Описать рекурсивную функцию  $\text{DigitCount}(S)$  целого типа, которая находит количество цифр в строке  $S$ , не используя оператор цикла. С помощью этой функции найти количество цифр в каждой из пяти данных строк.

**Recur13.** Описать рекурсивную функцию  $\text{Palindrom}(S)$  логического типа, возвращающую  $\text{True}$ , если строка  $S$  является *палиндромом* (то есть читается одинаково слева направо и справа налево), и  $\text{False}$  в противном случае. Оператор цикла в теле функции не использовать. Вывести значения функции  $\text{Palindrom}$  для пяти данных строк.

## 19.2 Разбор выражений

Во всех заданиях данного пункта предполагается, что исходные строки, определяющие выражения, не содержат пробелов. При выполнении заданий рекомендуется обойтись без операторов цикла.

**Recur14°.** Вывести значение целочисленного выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом:

$$\begin{aligned} \langle \text{выражение} \rangle &::= \langle \text{цифра} \rangle \mid \langle \text{выражение} \rangle + \langle \text{цифра} \rangle \mid \\ &\quad \langle \text{выражение} \rangle - \langle \text{цифра} \rangle \end{aligned}$$

**Recur15°.** Вывести значение целочисленного выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом:

$$\begin{aligned} \langle \text{выражение} \rangle &::= \langle \text{терм} \rangle \mid \langle \text{выражение} \rangle + \langle \text{терм} \rangle \mid \\ &\quad \langle \text{выражение} \rangle - \langle \text{терм} \rangle \\ \langle \text{терм} \rangle &::= \langle \text{цифра} \rangle \mid \langle \text{терм} \rangle * \langle \text{цифра} \rangle \end{aligned}$$

**Recur16°.** Вывести значение целочисленного выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом:

$$\begin{aligned} \langle \text{выражение} \rangle &::= \langle \text{терм} \rangle \mid \langle \text{выражение} \rangle + \langle \text{терм} \rangle \mid \\ &\quad \langle \text{выражение} \rangle - \langle \text{терм} \rangle \\ \langle \text{терм} \rangle &::= \langle \text{элемент} \rangle \mid \langle \text{терм} \rangle * \langle \text{элемент} \rangle \\ \langle \text{элемент} \rangle &::= \langle \text{цифра} \rangle \mid (\langle \text{выражение} \rangle) \end{aligned}$$

**Recur17.** Вывести значение целочисленного выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом:

$$\begin{aligned} \langle \text{выражение} \rangle &::= \langle \text{цифра} \rangle \mid \\ &\quad (\langle \text{выражение} \rangle \langle \text{знак} \rangle \langle \text{выражение} \rangle) \\ \langle \text{знак} \rangle &::= + \mid - \mid * \end{aligned}$$

**Recur18°.** Проверить правильность выражения, заданного в виде непустой строки  $S$  (выражение определяется по тем же правилам, что и в задании Recur17). Если выражение составлено правильно, то вывести  $\text{True}$ , иначе вывести  $\text{False}$ .

**Recur19.** Проверить правильность выражения, заданного в виде непустой строки  $S$  (выражение определяется по тем же правилам, что и в задании Recur17). Если выражение составлено правильно, то вывести 0, в противном случае вывести номер первого ошибочного (или лишнего) символа в строке  $S$ .

**Recur20.** Вывести значение целочисленного выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом (функция  $M$  возвращает максимальный из своих параметров, а функция  $m$  — минимальный):

$$\begin{aligned} \langle \text{выражение} \rangle &::= \langle \text{цифра} \rangle \mid M(\langle \text{выражение} \rangle, \langle \text{выражение} \rangle) \mid \\ &\quad m(\langle \text{выражение} \rangle, \langle \text{выражение} \rangle) \end{aligned}$$

**Recur21.** Вывести значение логического выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом («Т» — True, «F» — False):

$$\begin{aligned} \langle \text{выражение} \rangle &::= T \mid F \mid \text{And}(\langle \text{выражение} \rangle, \langle \text{выражение} \rangle) \mid \\ &\quad \text{Or}(\langle \text{выражение} \rangle, \langle \text{выражение} \rangle) \end{aligned}$$

**Recur22.** Вывести значение целочисленного выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом (функция  $M$  возвращает максимальный из своих параметров, а функция  $m$  — минимальный):

$$\begin{aligned} \langle \text{выражение} \rangle &::= \langle \text{цифра} \rangle \mid M(\langle \text{параметры} \rangle) \mid m(\langle \text{параметры} \rangle) \\ \langle \text{параметры} \rangle &::= \langle \text{выражение} \rangle \mid \langle \text{выражение} \rangle, \langle \text{параметры} \rangle \end{aligned}$$

**Recur23.** Вывести значение логического выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом («Т» — True, «F» — False):

$$\begin{aligned} \langle \text{выражение} \rangle &::= T \mid F \mid \text{And}(\langle \text{параметры} \rangle) \mid \text{Or}(\langle \text{параметры} \rangle) \\ \langle \text{параметры} \rangle &::= \langle \text{выражение} \rangle \mid \langle \text{выражение} \rangle, \langle \text{параметры} \rangle \end{aligned}$$

**Recur24.** Вывести значение логического выражения, заданного в виде строки  $S$ . Выражение определяется следующим образом («Т» — True, «F» — False):

$$\begin{aligned} \langle \text{выражение} \rangle &::= T \mid F \mid \text{And}(\langle \text{параметры} \rangle) \mid \\ &\quad \text{Or}(\langle \text{параметры} \rangle) \mid \text{Not}(\langle \text{выражение} \rangle) \\ \langle \text{параметры} \rangle &::= \langle \text{выражение} \rangle \mid \langle \text{выражение} \rangle, \langle \text{параметры} \rangle \end{aligned}$$

### 19.3 Перебор с возвратом

**Recur25°.** Дано дерево глубины  $N$ , каждая внутренняя вершина которого имеет  $K$  ( $< 10$ ) непосредственных потомков (нумеруются от 1 до  $K$ ). Корень дерева имеет номер 0. Записать в текстовый файл с данным именем все возможные пути, ведущие от корня к листьям. Перебирать пути, начиная с «самого левого» и заканчивая «самым правым» (при этом первыми заменять конечные элементы пути).

- Recur26.** Дано дерево глубины  $N$ , каждая внутренняя вершина которого имеет  $K$  ( $< 10$ ) непосредственных потомков (нумеруются от 1 до  $K$ ). Корень дерева имеет номер 0. Записать в текстовый файл с данным именем все пути, ведущие от корня к листьям и удовлетворяющие следующему условию: никакие соседние элементы пути не нумеруются одной и той же цифрой. Порядок перебора путей такой же, как в задании Recur25.
- Recur27.** Дано дерево глубины  $N$  ( $N$  — четное), каждая внутренняя вершина которого имеет 2 непосредственных потомка:  $A$  с весом 1 и  $B$  с весом  $-1$ . Корень дерева  $C$  имеет вес 0. Записать в текстовый файл с данным именем все пути от корня к листьям, удовлетворяющие следующему условию: суммарный вес элементов пути равен 0. Порядок перебора путей такой же, как в задании Recur25.
- Recur28.** Дано дерево глубины  $N$  того же типа, что и в задании Recur27. Записать в текстовый файл с данным именем все пути от корня к листьям, удовлетворяющие следующему условию: суммарный вес элементов для любого начального отрезка пути неотрицателен. Порядок перебора путей такой же, как в задании Recur25.
- Recur29.** Дано дерево глубины  $N$ , каждая внутренняя вершина которого имеет 3 непосредственных потомка:  $A$  с весом 1,  $B$  с весом 0 и  $C$  с весом  $-1$ . Корень дерева  $D$  имеет вес 0. Записать в текстовый файл с данным именем все пути от корня к листьям, удовлетворяющие следующим условиям: суммарный вес элементов для любого начального отрезка пути неположителен, а суммарный вес всех элементов пути равен 0. Порядок перебора путей такой же, как в задании Recur25.
- Recur30.** Дано дерево глубины  $N$  того же типа, что и в задании Recur29. Записать в текстовый файл с данным именем все пути от корня к листьям, удовлетворяющие следующим условиям: никакие соседние элементы пути не обозначаются одной и той же буквой, а суммарный вес всех элементов пути равен 0. Порядок перебора путей такой же, как в задании Recur25.

## 20 Указатели и динамические структуры данных: группа Pointer

Все числа, упоминаемые в заданиях данной группы, являются *целыми*. Все указатели имеют тип PNode и указывают на записи типа TNode.

На языке Pascal типы PNode и TNode описываются следующим образом:

```
type
  PNode = ^TNode;
  TNode = record
    Data: Integer;
    Next: PNode;
```