Составить программу-функцию вычисления факториала целого неотрицательного числа n.

Способ 1

Решение. Для целых неотрицательных чисел n факториал n обозначается через n! и определяется так:

$$n!=\begin{cases}1 & , \textit{если} \quad n=0 \ \text{ или } \ n=1;\\ 1\cdot 2\cdot \ldots \cdot n & , \textit{если} \quad n\ge 2.\end{cases}$$

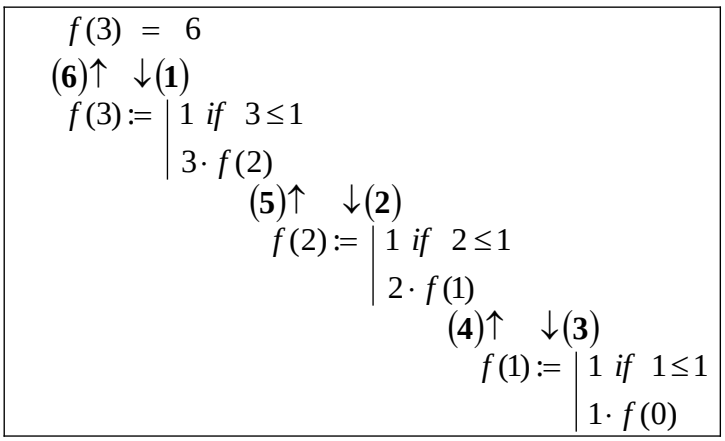
Случаи, для которых задача решается без рекурсивных вызовов, очевидны: f(0)=1, f(1)=1. Они и составляют базу рекурсии. Декомпозиция по параметру n реализуется по формуле: f(n) = n·f(n–1) (n = 1, 2, …). Поэтому вычисления f(n) можно организовать так:

```
program project1;
 var x:byte;
function f(n:byte):longint;
begin if n<=1then f:=1
else f:=n*f(n-1);
end;
begin  readln(x);writeln(f(x));
readln;
end.
```

Контрольные примеры.

$f(0)=1 \qquad f(5)=120 \qquad f(10)=3628800.$

Понять процесс реализации рекурсивных вызовов, то есть декомпозицию, и возвратов управления при организации отложенных вычислений f(n) можно из схемы (n = 3). Там около стрелок в круглых скобках жирными цифрами указаны номера последовательных шагов вычислений: (**1**), (**2**), (**3**) – декомпозиция; (**4**), (**5**), (**6**) – отложенные вычисления.



Способ2. Чтобы уменьшить глубину рекурсии, можно функцию f(n) можно было бы определить и так.

$$f(n):=\begin{cases}1 & \textit{if} \quad n\le 1\\ n\cdot(n-1)\cdot f(n-2)\end{cases}$$

```
program project2;
 var x:byte;
```

```
function f(n:byte):longint;
begin if n<=1then f:=1
else f:=n*(n-1)*f(n-2);
end;
begin  readln(x);writeln(f(x));
readln;
end.
```

Рекурсивное решение очень простое, но оно неоптимально по быстродействию: компьютер выполняет лишнюю работу, повторно вычисляя уже найденные значения.

Какой же выход? Напрашивается такое решение — хранить все предыдущие числа Фибоначчи в массиве.

Способ3 (рекурсивная функция+ массив)

```
Program Fibo; var I,N:integer;  D:array [1..50] of longint;
Function F(X:integer):longint;
begin
 if D[X]=0 then   if (X=1) or (X=2) then D[X]:=1 else D[X]:=F(X-1)+F(X-2);
 F:=D[X]
end;
begin
 For N:=1 to 50 do D[N]:=0;
 Readln(N);  writeln(F(N))
end.
```

Способ4

```
const N = 10;
var F: array[1..N] of integer;
begin
F[1]:= 1; F[2]:= 1;
for i:= 3 to N do F[i]:= F[i-1] + F[i-2];writeln(f[n]);
end.
```