

```

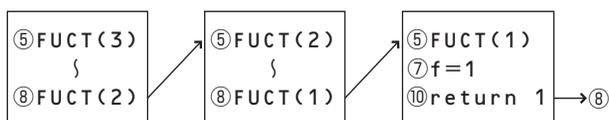
/* 主プログラム */
① ○整数型: n, f
② ・nの入力
③ ・f ← FUCT(n)
④ ・fの表示
/* 階乗の計算を行う関数 */
⑤ ○FUCT(整数型: n)
⑥ ↑n=1
⑦ | ・f←1      /* n=1のときは, 1!=1なので計算結果は1 */
  |-----|
⑧ | ・f←n×FUCT(n-1) /* 再帰呼出し */
  |-----|
⑨ ↓
⑩ return f

```

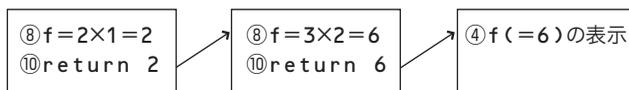
このプログラムで、3!を計算してみます。この結果は、6になるはずです。

まず、②でnに3が入力されます。次に、③で階乗の計算を行う関数FUCT()が呼び出されます。そして、⑥の判断でn≠1 (n=3) なので⑧で関数FUCT()が再帰的に呼び出されます。

次に、再帰呼出しの手順を示します。n=1になるまで、関数FUCT()が再帰的に呼び出されます。



次に、再帰呼出しから戻る手順を示します。関数FUCT()を呼び出した回数と同じだけ、return文が実行されます。そして、最後に主プログラムに戻ります。



●再帰の例

次のような再帰関数を考えてみます。

$$F(x) = x - 10 \quad (x > 100)$$

$$F(x) = F(F(x+11)) \quad (x \leq 100)$$

1

2

3

4

5

6

7

8

9

10

A



間違えやすい

プログラムの⑧におけるnの値は、関数FUCT()を呼び出す前の値であることに気をつける。同じプログラムを何回も呼び出してはいるが、別なプログラムを呼び出していると考えるとわかりやすい。呼び出している間、変数の値はスタックに保持されている。