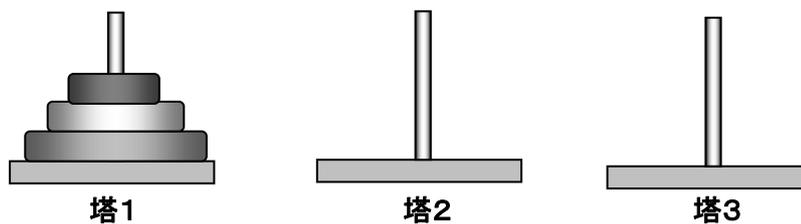


## 6. 再帰的プログラミング

### 6.1 ハノイの塔

再帰的プログラムとは、自分自身を呼び出すような関数です。「自分自身を呼び出す」というのは、イメージが湧きにくいので、自分自身のクローンを作って、そのクローンを呼び出すと考えましょう。再帰的プログラムの代表例は、ここに示すハノイの塔でしょう。

ハノイの塔とは、重なった円盤を 3 柱の間で移動する問題です。ただし、より小さい円盤が必ず上にくるように重ねます。たとえば、以下のような左の柱(塔 1)の 3 枚の円盤を右(塔 3)の柱に移します。



ハノイの塔の問題は、次のように再帰的な定義となります。

- ① N-1 枚の円盤を塔 1 から、塔 1 と塔 3 以外の塔 (塔 2) に移動
- ② 最も大きい円盤を塔 1 から塔 3 に移動
- ③ N-1 枚の円盤を塔 1 と塔 3 以外の塔 (塔 2) から、塔 3 に移動

x と y 以外の塔は  $6-x-y$  とすることができます。ハノイの塔の問題を解くプログラムを以下に示します。また、メインプログラム例も示しますので、動きを確認してみましょう。

```
void moveTo(int n, int x, int y) {
    printf("%n 第 %d 盤 : %d → %d ", n, x, y);
}
void hanoi(int n, int x, int y) {
    if(n>0) {
        hanoi(n-1, x, 6-x-y);
        moveTo(n, x, y);
        hanoi(n-1, 6-x-y, y);
    }
}
int _tmain(int argc, _TCHAR* argv[])
{
    int n;
    while (scanf("%d", &n) != EOF) { hanoi(n, 1, 3); printf("%n"); }
    return 0;
}
```