5.3 人目を気にして

プログラム [nthfib.cpp] は動作に問題はないけれど、ちょっとだけ気に入らない点がある。そこで、もう少しプログラムを追ってみよう。

プログラムは色々な入力に対して、“the **th Fibonacci . . . ” でフィボナッチ数を返してくれる。返されるフィボナッチ数は正確だが、**th が不自然になることがある。** が 1, 2, 3 のときは 1st, 2nd, 3rd になってくれないと困るのだ。もっとも 3 番までのフィボナッチ数は計算するまでもないけれど、ときに細かい点に注意しなければならないプログラムもある。これがその例とは言わないが、こんなときは switch 文で対応することができる。覚えておくとよいだろう。

programming list [nthfib2.cpp]

```
#include <iostream>

int fib(int n) {
    if(n > 2) {
        return (fib(n-1) + fib(n-2));
    } else {
        return 1;
    }
}

int main() {
    int num;
    std::cout << "input a number : "; std::cin >> num;
    switch(num % 10) {
        case 1:
            std::cout << "the " << num << "st Fibonacci number is "
                                        << fib(num) << std::endl; break;
        case 2:
            std::cout << "the " << num << "nd Fibonacci number is "
                                        << fib(num) << std::endl; break;
        case 3:
            std::cout << "the " << num << "rd Fibonacci number is "
                                        << fib(num) << std::endl; break;
        default: std::cout << "the " << num << "th Fibonacci number is "
                                        << fib(num) << std::endl;
    }

    return 0;
}
```

用法についてはプログラムを見れば分かると思う。switch() が受け取る num % 10 が、case のそれぞれの数に対応している。つまり、余りがそのまま case の分岐にあたるのである。余りを分岐の判断材料にしたことで、1, 2, 3 以外にも 21, 22, 23 などの数に対しても同じ効果が見込める

のだ。また、`cace 1 :` 等は字下げして `std::cout` を書いているが、`default :` は字下げをしていないことに大きな意味はない。`case` 文は字下げに無関係に書けることを示しただけだ。さらに、複文を書いても構わない。

また、各 `case` 文の最後に `break;` があることが重要だ。`break;` があるから、`case` が一致した処理をしたら、速やかに `switch(){}`から抜けられるのだ。もしどこかの `break;` がないと、`case` が一致したところから `break;` まで、または最後までの命令が実行されてしまう。例えば、`case 3 :` の最後に `break;` がないと、コンピュータは "the 3rd Fibonacci ..." と "the 3th Fibonacci ..." の 2 行を出力するだろう。

それにしても、プログラムの入力や出力を完璧にしようとすると、この例のようにプログラムサイズが大きくなってしまう。自分だけしか利用しないプログラムであれば、どの程度の値を入力するのか知っているし、出力が殺風景でも一向に構わないものだ。世に出回っているプログラムは、肝心の機能は一握りであっても、誰がどんな操作をしても困らないよう、用意周到に書かれているものだ。まったく、ご苦労なことである。

TRY! 実を言うと [`nthfib2.cpp`] はまだ完璧ではない。一見、英語の読みに対応して序数語尾が付けられるように見えるが、3 つだけ困った出力をする。11, 12, 13 のときだ。このままのプログラムでは "11st", "12nd", "13rd" と出力されてしまうが、"11th", "12th", "13th" が正しい。正しい出力になるよう、プログラムを修正せよ。