

プログラミング創造演習・2020 年度問題

解答期限：2020 年 5 月 6 日 (水) 17:00

C 言語のプログラムに関する以下の問題に答えよ。本や Web サイトで調べたりプログラムを書いて確かめたりしてもよいが、各自自力で解くこと。以下において、`int64_t` は 64 ビットの整数型とする。また整数型の値は 2 の補数で表されているとする。

1 巾乗の計算

以下の関数 `pow1` と `pow2` はどちらも整数 x の n 乗を計算する。 n が非負整数のとき、`pow1(x, n)` の実行における乗算の回数は明らかに n であり、計算時間もそれに比例する。一方 `pow2` はこれよりも高速に巾乗を計算する。

<pre>int64_t pow1(int64_t x, int n) { int64_t a = 1; while (n > 0) { a = a * x; n = n - 1; } return a; }</pre>	<pre>int64_t pow2(int64_t x, int n) { int64_t a = 1; while (n > 0) { if (n % 2 == 1) { a = a * x; n = n - 1; } else { x = x * x; n = n / 2; } } return a; }</pre>
---	--

問 1 `pow2(x, 50)` の実行における乗算の回数を答えよ。

問 2 n が非負整数のとき、`pow2(x, n)` の実行における乗算の回数を n の式で表せ。ここでは四則演算 ($+$, $-$, $*$, $/$) のほかに以下の関数を用いてよい。

- $\log_2(x)$: 2 を底とする x の対数 ($\log_2 x$)
- $\exp(x)$: 2 の x 乗 (2^x)
- $\text{floor}(x)$: x を超えない最大の整数 ($\lfloor x \rfloor$)
- $\text{ceil}(x)$: x より小さくない最小の整数 ($\lceil x \rceil$)
- $\text{abs}(x)$: x の絶対値 ($|x|$)
- $\text{bits}(x)$: x を 2 進数で表したときに 1 となる桁の数 (例: $\text{bits}(5)=2$)
- $\text{pos}(x)$: x が正のときに 1, それ以外のときに 0

なお、この問では整数演算のオーバーフローについては考えなくてよい。

問 3 `pow2(2, 63)` を計算したところ返値は負の数であった。その値を答えよ。

2 フィボナッチ数の計算

以下の関数 fib0, fib1, fib2 はいずれも n 番目のフィボナッチ数を計算する。fib2 は pow2 の考え方を用いて高速化をはかったものである。

```
int64_t fib0(int n) {
    if (n == 0) {
        return 0;
    }
    else if (n == 1) {
        return 1;
    }
    else {
        return fib0(n - 1)
            + fib0(n - 2);
    }
}
```

```
int64_t fib1(int n) {
    int64_t a = 0, b = 1;
    int64_t t;
    while (n > 0) {
        t = a;
        a = (1);
        b = (2);
        n = n - 1;
    }
    return a;
}
```

```
int64_t fib2(int n) {
    int64_t a = 0, b = 1;
    int64_t x = 0, y = 1;
    int64_t t;
    while (n > 0) {
        if (n % 2 == 1) {
            t = a;
            a = (3);
            b = (4);
            n = n - 1;
        }
        else {
            t = x;
            x = x * x + y * y;
            y = t * y + y * (t + y);
            n = n / 2;
        }
    }
    return a;
}
```

問 4 fib0(91) の返値となるべき数を答えよ。

問 5 fib0 のように再帰的に定義された関数では、関数の呼び出し回数を実行時間の目安とすることができる。例えば fib0(5) の実行における fib0 の呼び出し回数は 15 となる。1 秒間に関数呼び出しを 1 億回実行できるコンピュータ上での fib0(91) の実行時間はおおよそどのくらいか。関数呼び出し以外の計算時間は無視してよい。

問 6 関数 fib1 が n 番目のフィボナッチ数を計算するよう、空欄 (1) および (2) に入る式を答えよ。空欄に入る式は副作用を持たないものとする。

問 7 関数 fib2 が n 番目のフィボナッチ数を計算するよう、空欄 (3) および (4) に入る式を答えよ。空欄に入る式は副作用を持たないものとする。