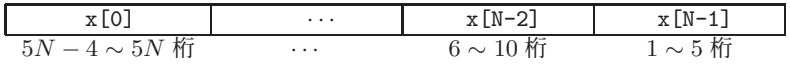
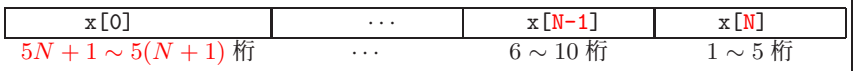
最終更新日: 2017 年 11 月 30 日

「C 言語による数値計算入門」(第 10〜13 刷) 正誤表

	誤	正
p.216, プログラム 10.1 に合わせて変更	下位桁からの桁上がりを up とし,  のように $5N$ 桁分の整数が格納されているとします.	下位桁からの桁上がりを up とし,  のように $5(N+1)$ 桁分の整数が格納されているとします.
p.216, 加算アルゴリズム	`For i = N-1, N-2, ..., 1, 0` `  sum = x[i] + y[i] + up`	`For i = N, N-1, ..., 1, 0` `  sum = x[i] + y[i] + up`
p.216, 減算アルゴリズム	`For i = N-1, N-2, ..., 1, 0` `  sub = x[i] - y[i] - borrow`	`For i = N, N-1, ..., 1, 0` `  sub = x[i] - y[i] - borrow`
p.217	ここでは簡単のため, 整数 $N(1 \sim 9999$ 程度$)$ と $x = \sum_{i=0}^{n} x[i] \cdot 100000^i$ との乗算を考えます. このとき, $$x \times N = \sum_{i=0}^{n} x[i] \cdot 100000^i \cdot N = \sum_{i=0}^{n} (x[i] \cdot N) \cdot 100000^i$$ となります. $(x[i] \cdot N)$ の計算においては下位からの桁上がりが発生する可能性があるので, あらかじめこの部分を $(x[i] \cdot N + up)$ と表して, $$x[i] \cdot N + up = \alpha \cdot 100000 + \beta$$ とおきます. このとき, α は桁上がりで, $x[i] \cdot N + up$ を 100000 で割ったときの商に対応します. また, β は $x[i] \cdot N + up$ を 100000 で割ったときの余りに対応し, これを $z[i]$ に格納します.	ここでは簡単のため, 整数 $n(1 \sim 9999$ 程度$)$ と $x = \sum_{i=0}^{N} x[i] \cdot 100000^{N-i}$ との乗算を考えます. このとき, $$x \times n = \sum_{i=0}^{N} x[i] \cdot 100000^{N-i} \cdot n = \sum_{i=0}^{N} (x[i] \cdot n) \cdot 100000^{N-i}$$ となります. $(x[i] \cdot n)$ の計算においては下位からの桁上がりが発生する可能性があるので, あらかじめこの部分を $(x[i] \cdot n + up)$ と表して, $$x[i] \cdot n + up = \alpha \cdot 100000 + \beta$$ とおきます. このとき, α は桁上がりで, $x[i] \cdot n + up$ を 100000 で割ったときの商に対応します. また, β は $x[i] \cdot n + up$ を 100000 で割ったときの余りに対応し, これを $z[i]$ に格納します.

	誤	正
p.217, 乗算アルゴリズム	For i = N-1, N-2, ..., 1, 0 prod = x[i]· N + up	For i = N, N-1, ..., 1, 0 prod = x[i]· n + up
p.217	乗算のときと同様に，整数 N と $x = \sum_{i=0}^{n} x[i] \cdot 100000^i$ との除算を考えます．除算は乗算と異なり，単純に商と余りだけを保持すれば	乗算のときと同様に，整数 n と $x = \sum_{i=0}^{N} x[i] \cdot 100000^{N-i}$ との除算を考えます．除算は乗算と異なり，単純に商と余りだけを保持すれば
p.217, 除算アルゴリズム	For i = 0, 1, ..., N-2, N-1 bunshi = amari · 100000 + x[i] z[i] = bunshi/N ＜－－－商の計算 amari = mod(bunshi, N) ＜－－－余りの計算	For i = 0, 1, ..., N-1, N bunshi = amari · 100000 + x[i] z[i] = bunshi/n ＜－－－商の計算 amari = mod(bunshi, n) ＜－－－余りの計算