
補足資料

人工知能パートナーシステム(AIPS)を支える

デジタル回路の世界

(ISBN978-4-88359-339-2)

著者名 萩原良昭 (hagihara-yoshiaki@aiplab.com)

<http://www.aiplab.com>

発行所名 青山社 (info@seizansha.co.jp)

<http://www.seizansha.co.jp>

$\sqrt{2} = 1.41421356237\dots$ の計算についての解説

有理数 (Rational Number) とは？

有限な桁数の整数 N, M を使って、 $Y=M/N$ (分数) として 表すことができる数を無理数という。

無理数 (Irrational Number) とは？

有限な桁数の整数 N, M を使って、 $Y=M/N$ (分数) として 表すことができない数を無理数という。

●けた数が有限な小数や整数は有理数である。かならず M/N と書ける！

たとえば $0.25 = 25/100 = 1/4$ 、 $1/10 = 0.1$ など。。。

2乗したら N なる数を $\text{Root}(N) = \sqrt{N}$ と書く。

$$\sqrt{2} = 1.41421356237\dots = \text{無理数}$$

$$\sqrt{3} = 1.73205080756\dots = \text{無理数}$$

$$\sqrt{4} = 2 = \text{有理数} \quad !$$

無理数(Irrational Number)とは？

有限な桁数の整数 N, M を使って、 $Y=M/N$ (分数) として 表すことができない数を無理数という。

$$\text{If } x^2 = 9, \quad x = 3 \text{ or } -3 .$$

$$\text{If } x^2 = a, \quad x = \sqrt{a} \text{ or } -\sqrt{a} .$$

無理数(Irrational Number)とは？

有限な桁数の整数 N, M を使って、 $Y=M/N$ (分数) として 表すことができない数を無理数という。

$$a^m = (a \text{ の } m \text{ 乗})$$

$$a^{1/n} = (a \text{ の } n \text{ 乗根}) = \sqrt[n]{a}$$

$$a^{m/n} = \sqrt[n]{a^m} = (a \text{ の } m \text{ 乗の } n \text{ 乗根})$$

無理数(Irrational Number)とは？

有限な桁数の整数 N, M を使って、 $Y=M/N$ (分数) として 表すことができない数を無理数という。

$a \geq 0$ $b \geq 0$ の時、

$$\sqrt{ab} = \sqrt{a} \cdot \sqrt{b}$$

$a \geq 0$ $b > 0$ の時、

$$\sqrt{\frac{a}{b}} = \frac{\sqrt{a}}{\sqrt{b}} \quad (b \neq 0)$$

● $\text{Root}(2) = 1.41421356\dots$ が
無理数であることを証明せよ。

● $\sqrt{2} = 1.41421356\dots$ が無理数であることを証明せよ。

無理数とは？

有限な桁数の整数 N, M を使って、

$Y = M/N$ として 表すことができない数を無理数という。

無理数でないとするとは？

● $\sqrt{2} = 1.41421356\dots$ が無理数であることを証明せよ。

無理数とは？

有限な桁数の整数 N, M を使って、

$Y = M/N$ として 表すことができない数を無理数という。

無理数でないとする？有理数である……

● $\sqrt{2} = 1.41421356\dots$ が無理数であることを証明せよ。

無理数とは？

有限な桁数の整数 N, M を使って、

$Y = M/N$ として 表すことができない数を無理数という。

無理数でないとする？有理数である……

有理数とは？ $\sqrt{2} = M/N$ と書けることになる。

有限な桁数の整数 N, M を使って、

$Y = M/N$ として 表すことができる数を有理数という。

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

● $\text{Root}(2)=1.41421356\dots$ が無理数であることを証明せよ。

無理数とは？

有限な桁数の整数 N, M を使って、

$Y=M/N$ として 表すことができない数を無理数という。

無理数でないとする？有理数である……

有理数とは？ $\sqrt{2} = M/N$ と書けることになる。

有限な桁数の整数 N, M を使って、

$Y=M/N$ として 表すことができる数を有理数という。

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

● $\sqrt{2} = 1.41421356\dots$ が無理数であることを証明せよ。

無理数でないとする？有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

● $\text{Root}(2)=1.41421356\dots$ が無理数であることを証明せよ。

無理数でないとする？有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$2 = (M \times M) / (N \times N)$$

$$2 \times N \times N = M \times M \quad \text{となる。}$$

● $\text{Root}(2)=1.41421356\dots$ が無理数であることを証明せよ。

無理数でないとする？有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$2 \times N \times N = M \times M \quad \text{となる。}$$

● $\sqrt{2}=1.41421356\dots$ が無理数であることを証明せよ。

無理数でないとする？有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$M = 2 \times M' \quad \text{となる。}$$

$$2 \times N \times N = M \times M \quad \text{となる。}$$

● $\text{Root}(2)=1.41421356\dots$ が無理数であることを証明せよ。

$$N \times N = 2 \times M' \times M' \quad \text{となる。}$$

$$N = 2 \times N' \quad \text{となる。}$$

無理数でないとする？有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$2 \times N \times N = M \times M \quad \text{となる。}$$

$$M = 2 \times M' \quad \text{となる。}$$

● $\text{Root}(2)=1.41421356\dots$ が無理数であることを証明せよ。

●MとNはあらかじめ通分してあったが、両方に同じ整数の、整数倍になっている???

$$N \times N = 2 \times M' \times M' \quad \text{となる。}$$

$$N = 2 \times N' \quad \text{となる。}$$

無理数でないとする? 有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

●MとNはあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$2 \times N \times N = M \times M \quad \text{となる。}$$

$$M = 2 \times M' \quad \text{となる。}$$

● $\sqrt{2}=1.41421356\dots$ が無理数であることを証明せよ。

無理数でないとすると？有理数である……

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \quad \text{と書けることになる。}$$

● M と N はあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$2 \times N \times N = M \times M \quad \text{となる。}$$

$$M = 2 \times M' \quad \text{となる。}$$

$$N \times N = 2 \times M' \times M' \quad \text{となる。}$$

$$N = 2 \times N' \quad \text{となる。}$$

● M と N はあらかじめ通分してあったが、両方に同じ整数の、整数倍になっている???

(a) Root(2)が有理数でない数、すなわち 無理数であることを説明
します。約数や倍数の概念をしっかりと理解する必要があります。

まとめ

●Root(2)=1.41421356...が無理数であることを証明せよ。

●矛盾が生じた。

無理数でないとすると？有理数である……

← 仮定が間違っている！

有理数なら有限な桁数の整数 N, M を使って、

$$\sqrt{2} = M/N \text{ と書けることになる。}$$

●MとNはあらかじめ通分しておき、両方に同じ整数の、整数倍にはなっていない。

$$2 \times N \times N = M \times M = \text{偶数} \quad \text{となる。}$$

$$M = 2 \times M' \text{ となる。}$$

$$N \times N = 2 \times M' \times M'$$

$$N = 2 \times N' \text{ となる。}$$

M が 奇数なら M x M も奇数。
M が 偶数なら M x M は偶数。
M x M が偶数になるには
Mが偶数の時のみ！

●MとNはあらかじめ通分してあったが、両方に同じ整数の、整数倍になっている???

(a) $\sqrt{2}$ が有理数でない数、すなわち 無理数であることを説明
します。約数や倍数の概念をしっかりと理解する必要があります。

まとめ

● $\sqrt{2} = 1.41421356\dots$ が無理数であることを証明せよ。

有理数 = 必ず N/M と書けるもの(それ以上通分できない)、 N と M は 有限整数
無理数 = N と M は 有限整数で、 N/M と書けないもの

$\sqrt{2}$ が 無理数であることの証明:

もし $\sqrt{2} = N/M$ と書けるとすると、 $\sqrt{2} M = N$ $2 * M * M = N * N$

したがって $N * N$ は 2 の整数倍、2つの整数のどちらかが2の整数倍のはず、
したがって、 $N = 2K$ と書ける。。 $2 * M * M = 2K * 2K = 4 * K * K$

したがって $M * M = 2 * K * K$ となる。

同様の論理で、 $M = 2 * H$ と書ける。。
かつ、 $N = 2K$ 。。。

M が 奇数なら $M * M$ も奇数。
 M が 偶数なら $M * M$ は偶数。
 $M * M$ が偶数になるには
 M が偶数の時のみ!

これは 最初の仮定 「 N/M は これ以上通分できない」と矛盾する。

したがって、 $\sqrt{2}$ は 有理数でありえない。

- (a) $\sqrt{2}$ が有理数でない数、すなわち 無理数であることを説明します。約数や倍数の概念をしっかりと理解する必要があります。
- (b) 次に $\sqrt{2}$ の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

(a) $\sqrt{2}$ が有理数でない数、すなわち 無理数であることを説明します。約数や倍数の概念をしっかりと理解する必要があります。

(b) 次に $\sqrt{2}$ の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 $Root(2)$ が 小数点以下 $i=8$ けたまで 求まっているとします。

$$Root(2) = 1.41421356 = 141421356 / 10^8$$

$$Root(2) = A(i) / 10^i \quad \text{where } i = 8 \text{ and } A(8) = 141421356$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithm
を説明し、多数あるその要素Program(Function)を紹介します。

今 $Root(2)$ が 小数点以下 $i=8$ けたまで 求まっているとします。

$$Root(2) = 1.41421356 = 141421356 / 10^8$$

$$Root(2) = A(i) / 10^i \quad \text{where } i = 8 \text{ and } A(8) = 141421356$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 $\text{Root}(2)$ が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 $\text{Root}(2)$ が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。}$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 $\text{Root}(2)$ が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。} X(9) \text{ を求めたい。}$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 $\text{Root}(2)$ が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。} X(9) \text{ を求めたい。}$$

$X(9)$ の値は 0 から 9 までの 数字である！

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 Root(2) が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。} X(9) \text{ を求めたい。}$$

$X(9)$ の値は 0 から 9 までの 数字である！

$$\text{Root}(2) = A(9) / 10^9 = A(8) / 10^8 + X(9) / 10^9$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 Root(2) が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。} X(9) \text{ を求めたい。}$$

$X(9)$ の値は 0 から 9 までの 数字である！

$$\text{Root}(2) = A(9) / 10^9 = A(8) / 10^8 + X(9) / 10^9$$

$$A(8) / 10^8 < A(8) / 10^8 + X(9) / 10^9 < \text{Root}(2) < A(8) / 10^8 + \{ X(9) + 1 \} / 10^9$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 Root(2) が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i = 8 \quad \text{and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。} X(9) \text{ を求めたい。}$$

$X(9)$ の値は 0 から 9 までの 数字である！

$$\text{Root}(2) = A(9) / 10^9 = A(8) / 10^8 + X(9) / 10^9$$

$$A(8) / 10^8 < A(8) / 10^8 + X(9) / 10^9 < \text{Root}(2) < A(8) / 10^8 + \{X(9)+1\} / 10^9$$

$$2 - A^2(i) / 10^{2i} = D(i) / 10^{2i} \quad \text{と定義した。}$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

今 Root(2) が 小数点以下 $i=8$ けたまで 求まっているとします。

$$\text{Root}(2) = 1.41421356 = 141421356 / 10^8$$

$$\text{Root}(2) = A(i) / 10^i \quad \text{where } i=8 \text{ and } A(8) = 141421356$$

つぎに $i=9$ $A(9)$ の値を求めたい。

$$A(9) = 10 * A(8) + X(9) \quad \text{と書く。} X(9) \text{ を求めたい。}$$

$X(9)$ の値は 0 から 9 までの 数字である！

$$\text{Root}(2) = A(9) / 10^9 = A(8) / 10^8 + X(9) / 10^9$$

$$A(8) / 10^8 < A(8) / 10^8 + X(9) / 10^9 < \text{Root}(2) < A(8) / 10^8 + \{X(9)+1\} / 10^9$$

$$2 - A^2(i) / 10^{2i} = D(i) / 10^{2i} \quad \text{と定義した。}$$

$$2 - A^2(i-1) / 10^{2i-2} = D(i-1) / 10^{2i-2} \quad \text{とやう。}$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

$$A(8)/10^8 < A(8)/10^8 + X(9)/10^9 < \text{Root}(2) < A(8)/10^8 + \{X(9)+1\}/10^9$$

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{とやう.}$$

(b) 次にRoot(2)の値を1万～10万ケタ計算するProgramのAlgorithmを説明し、多数あるその要素Program(Function)を紹介します。

$$A(i) = 10 * A(i-1) + X(i)$$

$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$A(8)/10^8 < A(8)/10^8 + X(9)/10^9 < \text{Root}(2) < A(8)/10^8 + \{X(9)+1\}/10^9$$

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{とす.}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると、} \quad 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{とす.}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると、} \quad 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{とすると.}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると、} \quad 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{とすると.}$$

$$A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると、} \quad 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$\begin{aligned}
 & 2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.} \\
 \rightarrow & \underline{2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{でも.}} \\
 & A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i} \\
 & = D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{でも...}
 \end{aligned}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると、} \quad 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$\begin{aligned}
 & 2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.} \\
 \rightarrow & \quad 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{も成り立つ.} \\
 \hline
 & A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i} \\
 & = D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{も成り立つ.}
 \end{aligned}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると、} \quad 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

$$A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i}$$

$$= D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

$$\text{従って, } (100)A^2(i-1) - A^2(i) = D(i) - (100)D(i-1) \quad \text{--- ①}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

$$\text{とすると, } 100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

$$A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i}$$

$$= D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{と書ける...}$$

従って、 $(100)A^2(i-1) - A^2(i) = D(i) - (100)D(i-1) \quad \text{--- ①}$

また、 $A(i) = (10)A(i-1) + X(i)$ と定義した。

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

とすると、 $100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

$$A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i}$$

$$= D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{と書ける...}$$

従って、 $(100)A^2(i-1) - A^2(i) = D(i) - (100)D(i-1)$ ①

また、 $A(i) = (10)A(i-1) + X(i)$ と定義した.

$$A^2(i) = (100)A^2(i-1) + (20)A(i-1)X(i) + X^2(i)$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

とすると、 $100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

$$A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i}$$

$$= D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

従って、 $(100)A^2(i-1) - A^2(i) = D(i) - (100)D(i-1)$ ①

また、 $A(i) = (10)A(i-1) + X(i)$ と定義した。

$$A^2(i) = (100)A^2(i-1) + (20)A(i-1)X(i) + X^2(i)$$

$$(100)A^2(i-1) - A^2(i) = -[(20)A(i-1) + X(i)]X(i) \quad \text{②}$$

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

とすると、 $100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$2 - A^2(i)/10^{2i} = D(i)/10^{2i} \quad \text{と定義した.}$$

$$\rightarrow 2 - A^2(i-1)/10^{2i-2} = D(i-1)/10^{2i-2} \quad \text{と書ける.}$$

$$A^2(i-1)/10^{2i-2} - A^2(i)/10^{2i}$$

$$= D(i)/10^{2i} - D(i-1)/10^{2i-2} \quad \text{と書ける...}$$

従って、 $(100)A^2(i-1) - A^2(i) = D(i) - (100)D(i-1)$ ①

また、 $A(i) = (10)A(i-1) + X(i)$ と定義した.

$$A^2(i) = (100)A^2(i-1) + (20)A(i-1)X(i) + X^2(i)$$

$$(100)A^2(i-1) - A^2(i) = -[(20)A(i-1) + X(i)]X(i) \quad \text{②}$$

①と②より、 $D(i) = (100)D(i-1) - X(i)[20A(i-1) + X(i)]$

$$A(i) = 10A(i-1) + X(i)$$

$$2 - A(i)^2/10^{2i} = D(i)/10^{2i} > 0$$

とすると、 $100D(i-1) - X(i)\{20A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$A(i) = 10 \cdot A(i-1) + X(i)$$

$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 \cdot D(i-1) - X(i) \{20 \cdot A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

$$A(i) = 10 * A(i-1) + X(i)$$

$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

X(i) は 0~9 の数字の中で D(i) > 0 の条件を満足する最大値である。

$$A(i) = 10 * A(i-1) + X(i)$$

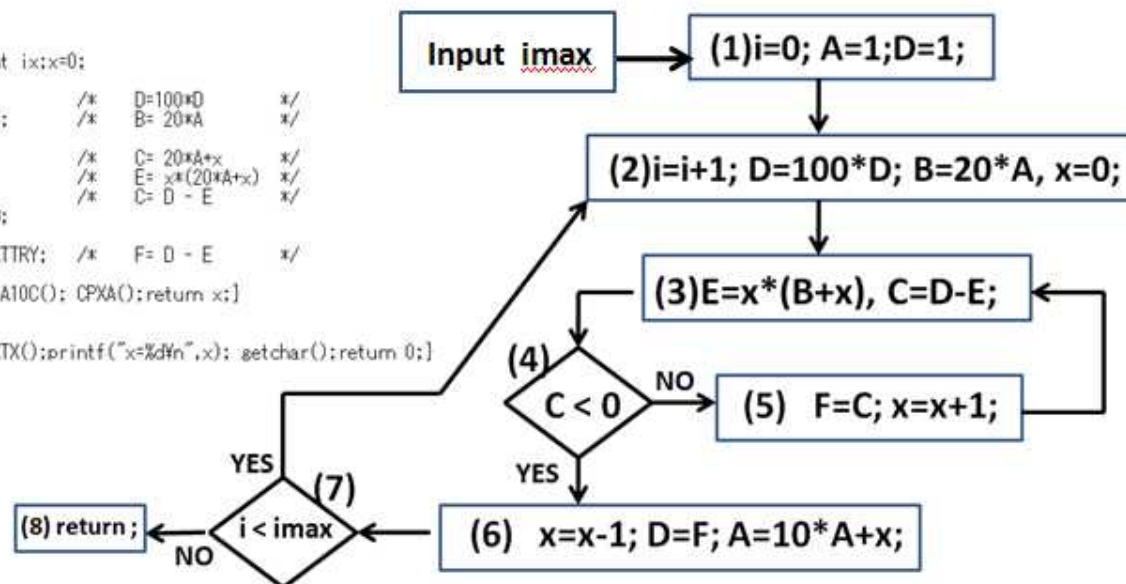
$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

X(i) は 0~9 の数字の中で D(i) > 0 の条件を満足する最大値である。

```
#include <stdio.h>
int x
int GETNEXTX(void){int ix:=0;
D100C():CT00(): /* D=100*D */
A100C():CT0E(): CPEB(): /* B= 20*A */
NEXTTRY:BPXC(): /* C= 20*A+x */
CXE(): /* E= x*(20*A+x) */
t=DSEC(): /* C= D - E */
if (t < 0) goto XEND;
CTOF():x=x+1;goto NEXTTRY: /* F= D - E */
XEND: x=x-1; FT0D(): A10C(): CPXA():return x;}
int main(void){GETNEXTX();printf("x=%d\n",x); getchar();return 0;}
```



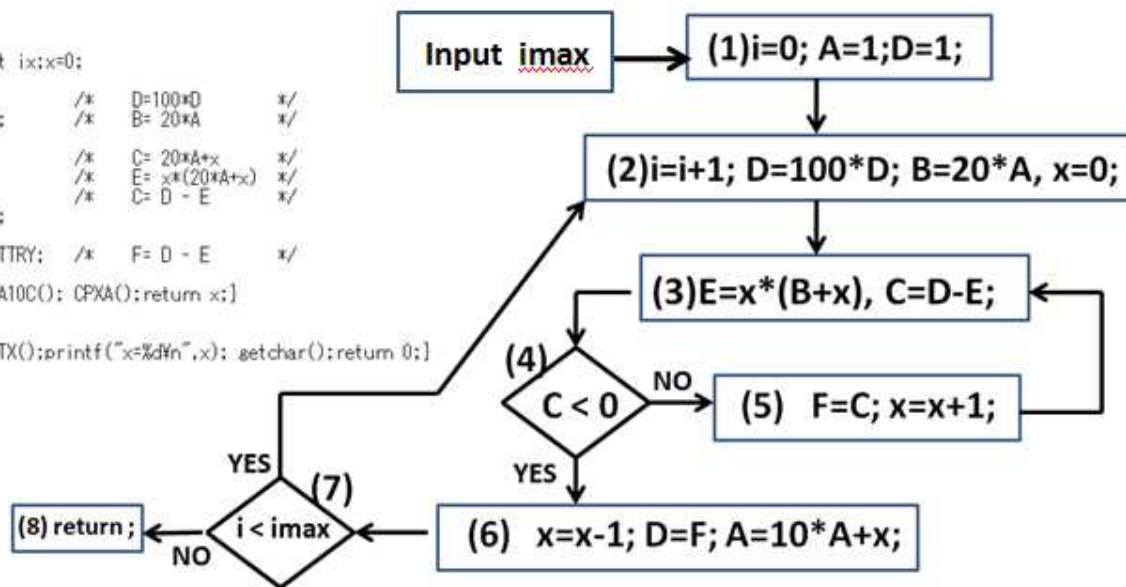
$$A(i) = 10 * A(i-1) + X(i)$$

$$2 - A(i) / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

```
#include <stdio.h>
int x
int GETNEXTX(void){int ix;x=0;
D100C():CT00(): /* D=100*D */
A10C():CT0E(): CPEB(): /* B= 20*A */
NEXTTRY:BPXC(): /* C= 20*A+x */
CXE(): /* E= x*(20*A+x) */
t=DSEC(): /* C= D - E */
if (t < 0) goto XEND;
CTOF():x=x+1;goto NEXTTRY; /* F= D - E */
XEND: x=x-1; FTOD(): A10C(): CPXA():return x;}
int main(void){GETNEXTX();printf("x=%d\n",x); setchar();return 0;}
```



/****** (Output Example of R.txt)*****

```
1.4142135623 7309504880 1688724209 6980785696 7187537694 8073176679 7379907324 7846210703 8850387534 3276415727
3501384623 0912297024 9248360558 5073721264 4121497099 9358314132 2266592750 5592755799 9505011527 8206057147
0109559971 6059702745 3459686201 4728517418 6408891986 0955232923 0484308714 3214508397 6260362799 5251407989
6872533965 4633180882 9640620615 2583523950 5474575028 7759961729 8355752203 3753185701 1354374603 4084988471
6038689997 0699004815 0305440277 9031645424 7823068492 9369186215 8057846311 1596688713 0130156185 6898723723
5288509264 8612494977 1542183342 0428568606 0146824720 7714358548 7415565706 9677653720 2264854470 1585880162
0758474922 6572260020 8558446652 1458398893 9443709265 9180031138 8246468157 0826301005 9485870400 3186480342
1948972782 9064104507 2636881313 7398552561 1732204024 5091227700 2269411275 7362728049 5738108967 5040183698
6836845072 5799364729 0607629969 4138047565 4823728997 1803268024 7442062926 9124859052 1810044598 4215059112
0249441341 7285314781 0580360337 1077309182 8693147101 7111168391 6581726889 4197587165 8215212822 9518488472
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

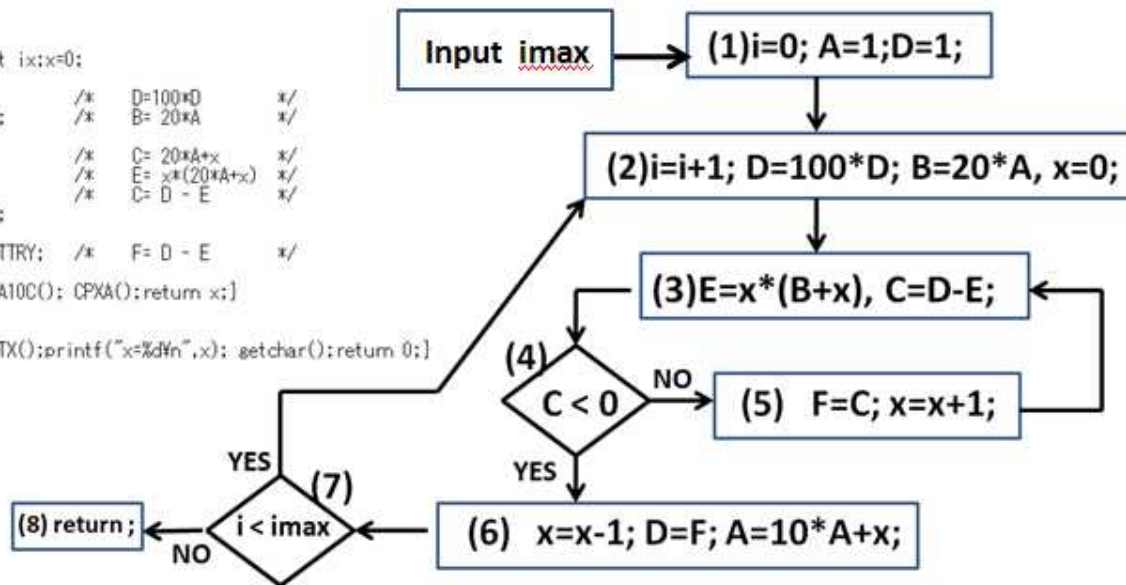
$$A(i) = 10 * A(i-1) + X(i)$$

$$2 - A(i) / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、 $100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$

という便利な関係を得る。

```
#include <stdio.h>
int x
int GETNEXTX(void){int ix;x=0;
D100C():CT00(): /* D=100*D */
A10C():CT0E(): CPEB(): /* B= 20*A */
NEXTTRY:BPXC(): /* C= 20*A+x */
CXE(): /* E= x*(20*A+x) */
t=DSEC(): /* C= D - E */
if (t < 0) goto XEND;
CTOF():x=x+1;goto NEXTTRY; /* F= D - E */
XEND: x=x-1; FTOD(): A10C(): CPXA():return x;}
int main(void){GETNEXTX();printf("x=%d\n",x); setchar();return 0;}
```



/****** (Output Example of R.txt)*****

```
1.4142135623 7309504880 1688724209 6980785696 7187537694 8073176679 7379907324 7846210703 8850387534 3276415727
3501384623 0912297024 9248360558 5073721264 4121497099 9358314132 2266592750 5592755799 9505011527 8206057147
0109559971 6059702745 3459686201 4728517418 6408891986 0955232923 0484308714 3214508397 6260362799 5251407989
6872533965 4633180882 9640620615 2583523950 5474575028 7759961729 8355752203 3753185701 1354374603 4084988471
6038689997 0699004815 0305440277 9031645424 7823068492 9369186215 8057846311 1596688713 0130156185 6898723723
5288509264 8612494977 1542183342 0428568606 0146824720 7714358548 7415565706 9677653720 2264854470 1585880162
0758474922 6572260020 8558446652 1458398893 9443709265 9180031138 8246468157 0826301005 9485870400 3186480342
1948972782 9064104507 2636881313 7398552561 1732204024 5091227700 2269411275 7362728049 5738108967 5040183698
6836845072 5799364729 0607629969 4138047565 4823728997 1803268024 7442062926 9124859052 1810044598 4215059112
0249441341 7285314781 0580360337 1077309182 8693147101 7111168391 6581726889 4197587165 8215212822 9518488472
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

```

#include <stdio.h>
int x;

int GETNEXTX(void){int ix;x=0;

D100C():CTOD(): /* D=100*D */
A10C():CTOE(): CPEB(): /* B= 20*A */

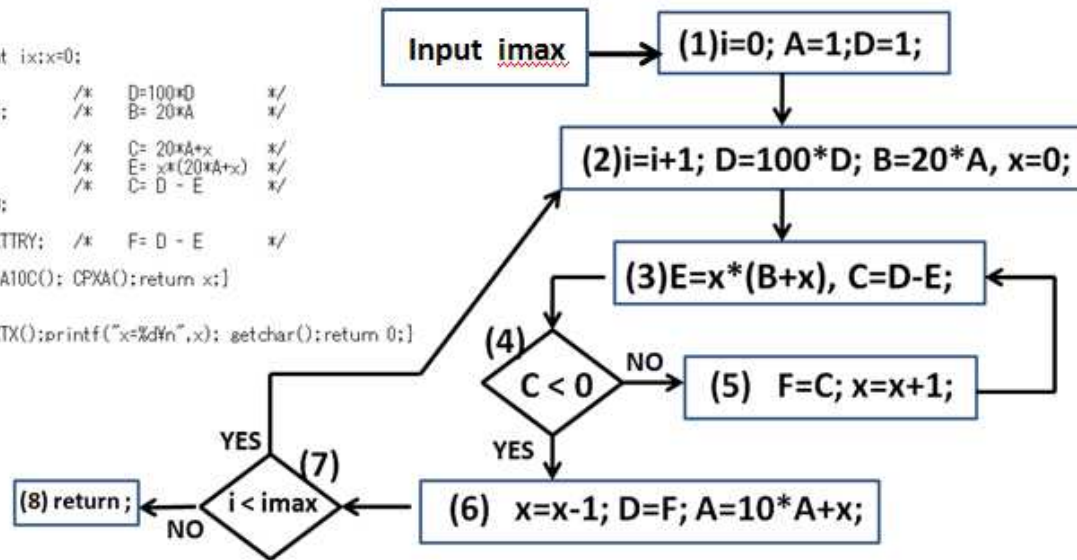
NEXTTRY:BPXC(): /* C= 20*A+x */
CXE(): /* E= x*(20*A+x) */
t=DSEC(): /* C= D - E */
if (t < 0 ) goto XEND;

CTOF():x=x+1;goto NEXTTRY; /* F= D - E */

XEND: x=x-1; FTOD(): A10C(): CPXA():return x;}

int main(void){GETNEXTX();printf("x=%d\n",x); getchar();return 0;}

```



***** (Output Example of R.txt)*****

```

1.4142135623 7309504880 1688724209 6980785696 7187537694 8073176679 7379907324 7846210703 8850387534 3276415727
3501384623 0912297024 9248360558 5073721264 4121497099 9358314132 2266592750 5592755799 9505011527 8206057147
0109559971 6059702745 3459686201 4728517418 6408891986 0955232923 0484308714 3214508397 6260362799 5251407989
6872533965 4633180882 9640620615 2583523950 5474575028 7759961729 8355752203 3753185701 1354374603 4084988471
6038689997 0699004815 0305440277 9031645424 7823068492 9369186215 8057846311 1596668713 0130156185 6898723723
5288509264 8612494977 1542183342 0428568606 0146824720 7714358548 7415565706 9677653720 2264854470 1585880162
0758474922 6572260020 8558446652 1458398893 9443709265 9180031138 8246468157 0826301005 9485870400 3186480342
1948972782 9064104507 2636881313 7398552561 1732204024 5091227700 2269411275 7362728049 5738108967 5040183698
6836845072 5799364729 0607629969 4138047565 4823728997 1803268024 7442062926 9124859052 1810044598 4215059112
0249441341 7285314781 0580360337 1077309182 8693147101 7111168391 6581726889 4197587165 8215212822 9518488472

```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Route(2) の次の桁の数 x をもとめる:

要素Program (Function)

$$A(i) = 10 * A(i-1) + X(i)$$

$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

とすると、

$$100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$$

という便利な関係を得る。

```
#include <stdio.h>

int x

int GETNEXTX(void){int ix;x=0;

D100C();CTOD();          /*  D=100*D      */
A10C();CTOE(); CPEB();    /*  B= 20*A     */

NEXTTRY:BPXC();          /*  C= 20*A+x   */
        CXE();           /*  E= x*(20*A+x) */
        t=DSEC();        /*  C= D - E    */
if (t < 0 ) goto XEND;

CTOF();x=x+1;goto NEXTTRY; /*  F= D - E    */

XEND: x=x-1; FTOD(); A10C(); CPXA();return x;}

int main(void){GETNEXTX();printf("x=%d\n",x); getchar();return 0;}
```

Function01(AONE).txt

Function02(DONE).txt

Function03(EZERO).txt

Function04(A10C).txt

Function05(D100C).txt

Function06(CTOA).txt

Function07(CTOD).txt

Function08(CTOE).txt

Function09(CTOF).txt

Function10(DTOE).txt

Function11(FTOD).txt

Function12(GTOE).txt

Function13(BPXC).txt

Function14(CPXA).txt

Function15(CPEB).txt

Function16(EPCG).txt

Function17(CXE).txt

Function18(DSEC).txt

Function19(GETNEXTX)...

Function20(Main).txt

$\sqrt{2}$ の計算に必要なProgram

●Root(2)の数値計算アルゴリズム(Algorithm)をまずしっかり理解する必要がある。。

その数値計算Algorithm を Flow Chart化して C-Program を完成させる。。

最終的に 自分のパソコンで Root(2)=1.41421356~の値を 1万桁求めたいが。。そう簡単でない！

$\sqrt{2} = 1.41421356 \dots$ を求めたい。

$$(\sqrt{2})^2 - (\text{近似値})^2 = (\text{2乗誤差})$$

$$100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} = D(i) > 0$$

$$2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

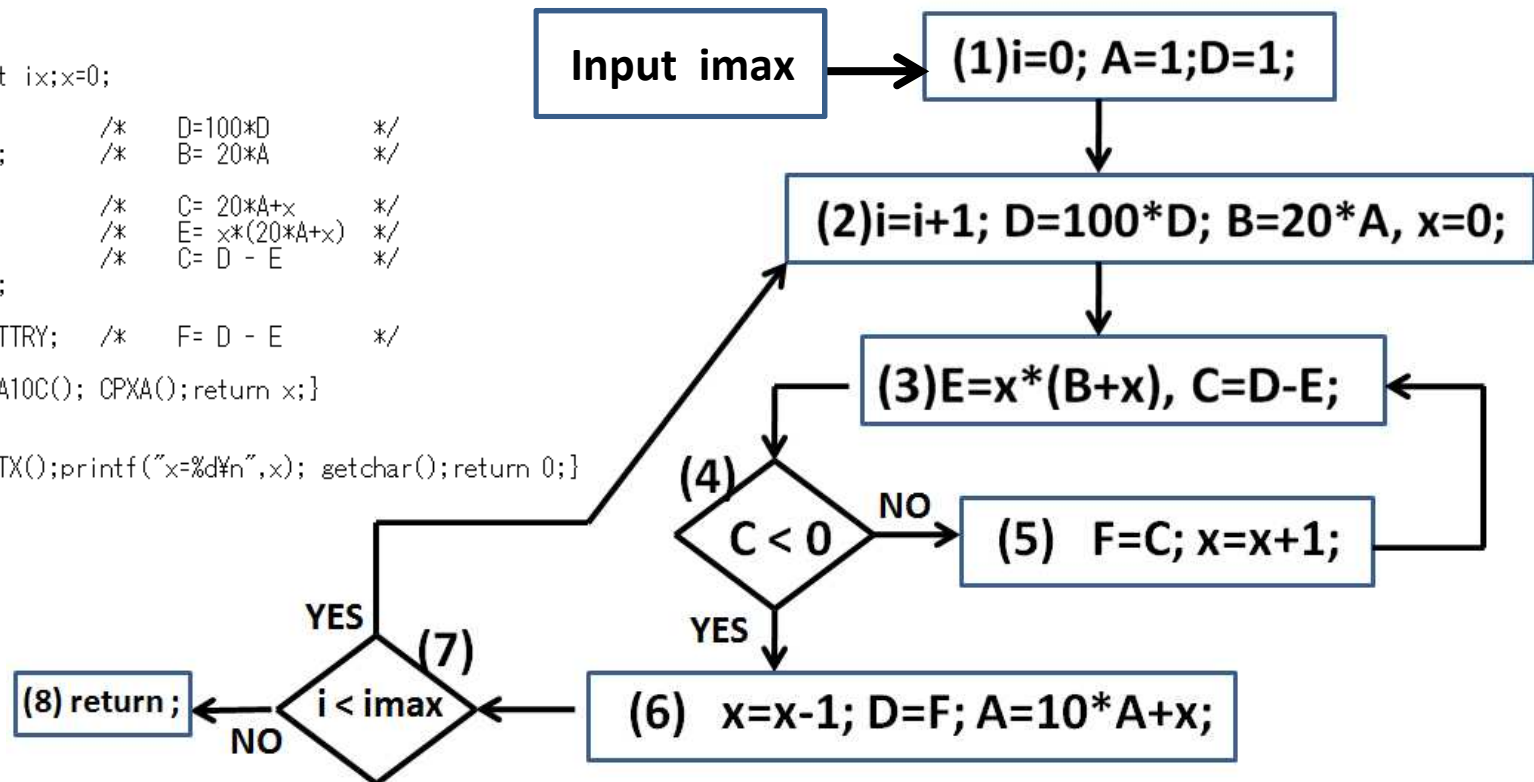
$$A(i) = 10 * A(i-1) + X(i)$$

X(i) の値を 0,1,2,3 と Tryして D(i) >0を満足する 最大の X(i)を求める。
X(i)が決まると D(i)も A(i)も もとまる。。。それを 繰り返せばよい！

```
#include <stdio.h>
int x

int GETNEXTX(void){int ix;x=0;
D100C();CTOD(); /* D=100*D */
A10C();CTOE(); CPEB(); /* B= 20*A */
NEXTTRY:BPXC(); /* C= 20*A+x */
CXE(); /* E= x*(20*A+x) */
t=DSEC(); /* C= D - E */
if (t < 0 ) goto XEND;
CTOF();x=x+1;goto NEXTTRY; /* F= D - E */
XEND: x=x-1; FTOD(); A10C(); CPXA();return x;}

int main(void){GETNEXTX();printf("x=%d\n",x); getchar();return 0;}
```



$\sqrt{2} = 1.41421356 \dots$ を求めたい。

近似値: $R(i) = A(i) / 10^i$ を求めたい。

		<u>i</u>	<u>X(i)</u>	<u>R(i)</u>
A(0) = 1	X(0) = 1	0000000	1	1
A(1) = 14	X(1) = 4	0000001	4	1.4
A(2) = 141	X(2) = 1	0000002	1	1.41
A(3) = 1414	X(3) = 4	0000003	4	1.414
A(4) = 14142	X(4) = 2	0000004	2	1.4142
A(5) = 141421	X(5) = 1	0000005	1	1.41421
A(6) = 1414213	X(6) = 3	0000006	3	1.414213
A(7) = 14142135	X(7) = 5	0000007	5	1.4142135
A(8) = 141421356	X(8) = 6	0000008	6	1.41421356
A(9) = -----		0000009	-	-
		0000010	-	-

希望 Output Format

$$(\sqrt{2})^2 - (\text{近似値})^2 = (\text{2乗誤差}) \quad 2 - A(i)^2 / 10^{2i} = D(i) / 10^{2i} > 0$$

$$A(i) = 10 * A(i-1) + X(i)$$

$$D(i) = 100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} > 0$$

という便利な関係に注目して 数値計算 Algorithm を構築する。

C言語の復習:

Function01(AONE):

File A (A.txt) に Data '1' を記載して 閉じる。
A=1 に 初期値設定する。

```
#include <stdio.h>

int AONE(void) {char j;

FILE *fpA; fpA=fopen("A.txt","w");

j='1';  fputc(j,fpA);

fclose(fpA); return 0;}

int main(void){AONE();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function02(DONE):

File D (D.txt) に Data '1' を記載して 閉じる。
D=1 に 初期値設定する。

```
#include <stdio.h>

int DONE(void) {char j;

FILE *fpD; fpD=fopen("D.txt","w");

j='1';  fputc(j,fpD);

fclose(fpD); return 0;}

int main(void){DONE();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function03(EZERO):

NUM02(SQRT2) が 最終の完成した Program だが。。。
まずは 単純な Program から 組み立ててみる。。。。

File E (E.txt) に Data'0' を 記載し、初期設定する。

E = 0 (右上位表示)

```
#include <stdio.h>

int EZERO(void){char j;FILE *fpE;fpE=fopen("E.txt","w");

                j='0';fputc(j,fpE);fclose(fpE);return 0;}

int main(void){EZERO();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function04(A10C):

File A (A.txt) の 値を 10倍して File C (C.txt) に 保存する。

実際には最右桁に0を追加して、
AのDataを右に1桁ずらす ← (右上位表示)
A=1234871 の場合、 C=01234871 となる。
実際のC の値は 通常の 左上位表示で書くと
17843210 を意味する。

```
#include <stdio.h> int A10C(void){ char j;FILE *fpA;FILE *fpC;
    fpA=fopen("A.txt","r");fpC=fopen("C.txt","w");
    j='0';fputc(j,fpC);
ONCEMORE: j=fgetc(fpA);
if ( j == '1' ) goto NEXTSTEP; if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP; if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP; if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP; if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP; if ( j == '0' ) goto NEXTSTEP;
fclose(fpA);fclose(fpC);return 0;
NEXTSTEP: fputc(j,fpC);goto ONCEMORE;}
int main(void){A10C();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function05(D100C):

File D (D.txt) の 値を 100倍して File C (C.txt) に 保存する。

実際には最右桁に00を追加して、
AのDataを右に2桁ずらす ← (右上位表示)
D=1234871 の場合、 C=001234871 となる。
実際のC の値は 通常の 左上位表示で書くと
178432100 を意味する。

```
#include <stdio.h>

int D100C(void){ char j;FILE *fpD;FILE *fpC;
                fpD=fopen("D.txt","r");fpC=fopen("C.txt","w");
j='0';fputc(j,fpC);fputc(j,fpC);
ONCEMORE: j=fgetc(fpD);

if ( j == '1' ) goto NEXTSTEP; if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP; if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP; if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP; if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP; if ( j == '0' ) goto NEXTSTEP;

fclose(fpD);fclose(fpC);return 0;
NEXTSTEP: fputc(j,fpC);goto ONCEMORE;}

int main(void){D100C();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function 06 (CTOA):

File C (C.txt) の Data を そのまま
File A (A.txt) に 保存する。

A = C (右上位表示)

```
#include <stdio.h>

int CTOA(void){char j;FILE *fpC;FILE *fpA;
                fpC=fopen("C.txt","r");fpA=fopen("A.txt","w");

ONCEMORE: j=fgetc(fpC);





















if ( j == '1' ) goto NEXTSTEP; if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP; if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP; if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP; if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP; if ( j == '0' ) goto NEXTSTEP;

fclose(fpC);fclose(fpA);return 0;

NEXTSTEP: fputc(j,fpA);goto ONCEMORE;}

int main(void){CTOA();return 0;}
```

要素Program (Function)

-  Function01(AONE).txt
-  Function02(DONE).txt
-  Function03(EZERO).txt
-  Function04(A10C).txt
-  Function05(D100C).txt
-  Function06(CTOA).txt
-  Function07(CTOD).txt
-  Function08(CTOE).txt
-  Function09(CTOF).txt
-  Function10(DTOE).txt
-  Function11(FTOD).txt
-  Function12(GTOE).txt
-  Function13(BPXC).txt
-  Function14(CPXA).txt
-  Function15(CPEB).txt
-  Function16(EPCG).txt
-  Function17(CXE).txt
-  Function18(DSEC).txt
-  Function19(GETNEXTX)...
-  Function20(Main).txt

Function07(CTOD):

File C (C.txt) の Data を
そのまま File D (D.txt) に 保存する。

D = C (右上位表示)

```
#include <stdio.h>

int CTOD(void){char j;FILE *fpC;FILE *fpD;
                fpC=fopen("C.txt","r");fpD=fopen("D.txt","w");
ONCEMORE: j=fgetc(fpC);

if ( j == '1' ) goto NEXTSTEP;if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP;if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP;if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP;if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP;if ( j == '0' ) goto NEXTSTEP;

fclose(fpC);fclose(fpD);return 0;
NEXTSTEP: fputc(j,fpD);goto ONCEMORE;}

int main(void){CTOD();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function08(CTOE):

File C (C.txt) の Data を
そのまま File E (E.txt) に 保存する。

E = C (右上位表示)

```
#include <stdio.h>

int CTOE(void){char j;FILE *fpC;FILE *fpE;
                fpC=fopen("C.txt","r");fpE=fopen("E.txt","w");

ONCEMORE: j=fgetc(fpC);





















if ( j == '1' ) goto NEXTSTEP; if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP; if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP; if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP; if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP; if ( j == '0' ) goto NEXTSTEP;

fclose(fpC);fclose(fpE);return 0;

NEXTSTEP: fputc(j,fpE);goto ONCEMORE;}

int main(void){CTOE();return 0;}
```

要素Program (Function)

-  Function01(AONE).txt
-  Function02(DONE).txt
-  Function03(EZERO).txt
-  Function04(A10C).txt
-  Function05(D100C).txt
-  Function06(CTOA).txt
-  Function07(CTOD).txt
-  Function08(CTOE).txt
-  Function09(CTOF).txt
-  Function10(DTOE).txt
-  Function11(FTOD).txt
-  Function12(GTOE).txt
-  Function13(BPXC).txt
-  Function14(CPXA).txt
-  Function15(CPEB).txt
-  Function16(EPCG).txt
-  Function17(CXE).txt
-  Function18(DSEC).txt
-  Function19(GETNEXTX)...
-  Function20(Main).txt

Function09(CTOF):

File C (C.txt) の Data を
そのまま File F (F.txt) に 保存する。

F = C (右上位表示)

```
#include <stdio.h>

int CTOF(void){char j;FILE *fpC;FILE *fpF;
                fpC=fopen("C.txt","r");fpF=fopen("F.txt","w");

ONCEMORE: j=fgetc(fpC);





















if ( j == '1' ) goto NEXTSTEP;if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP;if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP;if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP;if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP;if ( j == '0' ) goto NEXTSTEP;

fclose(fpC);fclose(fpF);return 0;

NEXTSTEP: fputc(j,fpF);goto ONCEMORE;}

int main(void){CTOF();return 0;}
```

要素Program (Function)

-  Function01(AONE).txt
-  Function02(DONE).txt
-  Function03(EZERO).txt
-  Function04(A10C).txt
-  Function05(D100C).txt
-  Function06(CTOA).txt
-  Function07(CTOD).txt
-  Function08(CTOE).txt
-  Function09(CTOF).txt
-  Function10(DTOE).txt
-  Function11(FTOD).txt
-  Function12(GTOE).txt
-  Function13(BPXC).txt
-  Function14(CPXA).txt
-  Function15(CPEB).txt
-  Function16(EPCG).txt
-  Function17(CXE).txt
-  Function18(DSEC).txt
-  Function19(GETNEXTX)...
-  Function20(Main).txt

Function10(DTOE):

File D (D.txt) の Data を
そのまま File E (E.txt) に 保存する。

E = D (右上位表示)

```
#include <stdio.h>

int DTOE(void){char j;FILE *fpD;FILE *fpE;
                fpD=fopen("D.txt","r");fpE=fopen("E.txt","w");

ONCEMORE: j=fgetc(fpD);

if ( j == '1' ) goto NEXTSTEP;if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP;if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP;if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP;if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP;if ( j == '0' ) goto NEXTSTEP;

fclose(fpD);fclose(fpE);return 0;
NEXTSTEP: fputc(j,fpE);goto ONCEMORE;}

int main(void){DTOE();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function11(FTOD):

File F (F.txt) の Data を
そのまま File D (D.txt) に 保存する。

D = F (右上位表示)

```
#include <stdio.h>

int FTOD(void){char j;FILE *fpF;FILE *fpD;
                fpF=fopen("F.txt","r");fpD=fopen("D.txt","w");

ONCEMORE: j=fgetc(fpF);

if ( j == '1' ) goto NEXTSTEP; if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP; if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP; if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP; if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP; if ( j == '0' ) goto NEXTSTEP;

fclose(fpF);fclose(fpD);return 0;

NEXTSTEP: fputc(j,fpD);goto ONCEMORE;}

int main(void){FTOD();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function 12 (GTOE):

File G (G.txt) の Data を
そのまま File E (E.txt) に 保存する。

E = G (右上位表示)

```
#include <stdio.h>

int GTOE(void){char j;FILE *fpG;FILE *fpE;
                fpG=fopen("G.txt","r");fpE=fopen("E.txt","w");
ONCEMORE: j=fgetc(fpG);
if ( j == '1' ) goto NEXTSTEP; if ( j == '2' ) goto NEXTSTEP;
if ( j == '3' ) goto NEXTSTEP; if ( j == '4' ) goto NEXTSTEP;
if ( j == '5' ) goto NEXTSTEP; if ( j == '6' ) goto NEXTSTEP;
if ( j == '7' ) goto NEXTSTEP; if ( j == '8' ) goto NEXTSTEP;
if ( j == '9' ) goto NEXTSTEP; if ( j == '0' ) goto NEXTSTEP;

fclose(fpG);fclose(fpE);return 0;

int main(void){GTOE();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Program13(BPXC):

File B (B.txt) の Data に X を加算して、
File C (C.txt) に 保存する。

C = B + X (右上位表示)

```
#include <stdio.h> int x=9;
int BPXC(void){ char j; int carry,jj,xx; FILE *fpB;FILE *fpC;
                fpB=fopen("B.txt","r");fpC=fopen("C.txt","w");
                carry=0;xx=x;
ONCEMORE: j=fgetc(fpB); jj=-1;
if ( j == '1' ) jj=1;if ( j == '2' ) jj=2;if ( j == '3' ) jj=3;
if ( j == '4' ) jj=4;if ( j == '5' ) jj=5;if ( j == '6' ) jj=6;
if ( j == '7' ) jj=7;if ( j == '8' ) jj=8;if ( j == '9' ) jj=9;
if ( j == '0' ) jj=0;
if ( jj < 0 ) goto STEP01;
jj=jj+carry+xx;xx=0;carry=0;if ( jj > 9 ) { carry =1 ; jj=jj-10; }
    if ( jj==0 ) j='0'; if ( jj==1 ) j='1'; if ( jj==2 ) j='2';
    if ( jj==3 ) j='3'; if ( jj==4 ) j='4'; if ( jj==5 ) j='5';
    if ( jj==6 ) j='6'; if ( jj==7 ) j='7'; if ( jj==8 ) j='8';
    if ( jj==9 ) j='9'; fputc(j,fpC);goto ONCEMORE;
STEP01: if ( carry == 0 ) goto STEP03;j='1'; fputc(j,fpC);
STEP03: fclose(fpB);fclose(fpC);return 0;}
int main(void){BPXC();return 0;}
```

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt

Function14(CPXA):

```
#include <stdio.h>
int x=9;
int CPXA(void){ char j; int carry,jj,xx; FILE *fpC;FILE *fpA;
                fpC=fopen("C.txt","r");fpA=fopen("A.txt","w");
carry=0;xx=x;
ONCEMORE: j=fgetc(fpC); jj=-1;
if ( j == '1' ) jj=1;if ( j == '2' ) jj=2;if ( j == '3' ) jj=3;
if ( j == '4' ) jj=4;if ( j == '5' ) jj=5;if ( j == '6' ) jj=6;
if ( j == '7' ) jj=7;if ( j == '8' ) jj=8;if ( j == '9' ) jj=9;
if ( j == '0' ) jj=0;
if ( jj < 0 ) goto STEP01;
jj=jj+carry+xx;xx=0;carry=0;if ( jj > 9 ) { carry =1 ; jj=jj-10; }
if ( jj==0 ) j='0'; if ( jj==1 ) j='1'; if ( jj==2 ) j='2';
if ( jj==3 ) j='3'; if ( jj==4 ) j='4'; if ( jj==5 ) j='5';
if ( jj==6 ) j='6'; if ( jj==7 ) j='7'; if ( jj==8 ) j='8';
if ( jj==9 ) j='9'; fputc(j,fpA);goto ONCEMORE;





















STEP01: if ( carry == 0 ) goto STEP03;j='1'; fputc(j,fpA);
STEP03: fclose(fpC);fclose(fpA);return 0;}

int main(void){CPXA();return 0;}
```

A = C + x (右上位表示)

**File C (C.txt) の Data に
x を加算して、File A(A.txt)
に 保存する。**

要素Program (Function)

-  Function01(AONE).txt
-  Function02(DONE).txt
-  Function03(EZERO).txt
-  Function04(A10C).txt
-  Function05(D100C).txt
-  Function06(CTOA).txt
-  Function07(CTOD).txt
-  Function08(CTOE).txt
-  Function09(CTOF).txt
-  Function10(DTOE).txt
-  Function11(FTOD).txt
-  Function12(GTOE).txt
-  Function13(BPXC).txt
-  Function14(CPXA).txt
-  Function15(CPEB).txt
-  Function16(EPCG).txt
-  Function17(CXE).txt
-  Function18(DSEC).txt
-  Function19(GETNEXTX)...
-  Function20(Main).txt

Function15(CPEB):

```
#include <stdio.h>

int CPEB(void){ char j,k; int carry,jj,kk;FILE *fpC;FILE *fpE;FILE *fpB;
                fpC=fopen("C.txt","r");fpE=fopen("E.txt","r");
                fpB=fopen("B.txt","w");carry=0;

ONCEMORE: j=fgetc(fpC); k=fgetc(fpE);kk=-1;jj=-1;

if ( j == '1' ) jj=1;if ( j == '2' ) jj=2;if ( j == '3' ) jj=3;
if ( j == '4' ) jj=4;if ( j == '5' ) jj=5;if ( j == '6' ) jj=6;
if ( j == '7' ) jj=7;if ( j == '8' ) jj=8;if ( j == '9' ) jj=9;
if ( j == '0' ) jj=0;

if ( k == '1' ) kk=1;if ( k == '2' ) kk=2;if ( k == '3' ) kk=3;
if ( k == '4' ) kk=4;if ( k == '5' ) kk=5;if ( k == '6' ) kk=6;
if ( k == '7' ) kk=7;if ( k == '8' ) kk=8;if ( k == '9' ) kk=9;
if ( k == '0' ) kk=0;

if ( kk < 0 ) goto STEP03;if ( jj < 0 ) goto STEP01;

STEP02:kk=kk+jj+carry;carry=0; if ( kk > 9 ) { carry =1 ; kk=kk-10; }

if ( kk==0 ) k='0'; if ( kk==1 ) k='1'; if ( kk==2 ) k='2';
if ( kk==3 ) k='3'; if ( kk==4 ) k='4'; if ( kk==5 ) k='5';
if ( kk==6 ) k='6'; if ( kk==7 ) k='7'; if ( kk==8 ) k='8';
if ( kk==9 ) k='9';

fputc(k,fpB);goto ONCEMORE;

STEP01: jj=0; goto STEP02;

STEP03:if ( jj < 0 ) goto STEP04;kk=0; goto STEP02; B = C + E
STEP04:if ( carry == 0 ) goto STEP05;k='1';fputc(k,fpB);
STEP05:fclose(fpC);fclose(fpE);fclose(fpB);return 0;}

int main(void){CPEB();return 0;} (右上位表示)
```

**File C (C.txt) の Data と File E (E.txt)
の Data を加算して、File B (B.txt)
に保存する。**

B = C + E

(右上位表示)

要素Program (Function)

- 📄 Function01(AONE).txt
- 📄 Function02(DONE).txt
- 📄 Function03(EZERO).txt
- 📄 Function04(A10C).txt
- 📄 Function05(D100C).txt
- 📄 Function06(CTOA).txt
- 📄 Function07(CTOD).txt
- 📄 Function08(CTOE).txt
- 📄 Function09(CTOF).txt
- 📄 Function10(DTOE).txt
- 📄 Function11(FTOD).txt
- 📄 Function12(GTOE).txt
- 📄 Function13(BPXC).txt
- 📄 Function14(CPXA).txt
- 📄 Function15(CPEB).txt
- 📄 Function16(EPCG).txt
- 📄 Function17(CXE).txt
- 📄 Function18(DSEC).txt
- 📄 Function19(GETNEXTX)...
- 📄 Function20(Main).txt

Function 16 (EPCG):

```
#include <stdio.h>

int EPCG(void){ char j,k; int carry,jj,kk;FILE *fpE;FILE *fpC;FILE *fpG;
                fpE=fopen("E.txt","r");fpC=fopen("C.txt","r");
                fpG=fopen("G.txt","w");carry=0;

ONCEMORE: j=fgetc(fpE); k=fgetc(fpC);kk=-1;jj=-1;

if ( j == '1' ) jj=1; if ( j == '2' ) jj=2; if ( j == '3' ) jj=3;
if ( j == '4' ) jj=4; if ( j == '5' ) jj=5; if ( j == '6' ) jj=6;
if ( j == '7' ) jj=7; if ( j == '8' ) jj=8; if ( j == '9' ) jj=9;
if ( j == '0' ) jj=0;

if ( k == '1' ) kk=1; if ( k == '2' ) kk=2; if ( k == '3' ) kk=3;
if ( k == '4' ) kk=4; if ( k == '5' ) kk=5; if ( k == '6' ) kk=6;
if ( k == '7' ) kk=7; if ( k == '8' ) kk=8; if ( k == '9' ) kk=9;
if ( k == '0' ) kk=0;

if ( kk < 0 ) goto STEP03;if ( jj < 0 ) goto STEP01;

STEP02:kk=kk+jj+carry;carry=0; if ( kk > 9 ) { carry =1 ; kk=kk-10; }

if ( kk==0 ) k='0'; if ( kk==1 ) k='1'; if ( kk==2 ) k='2';
if ( kk==3 ) k='3'; if ( kk==4 ) k='4'; if ( kk==5 ) k='5';
if ( kk==6 ) k='6'; if ( kk==7 ) k='7'; if ( kk==8 ) k='8';
if ( kk==9 ) k='9';

fputc(k,fpG);goto ONCEMORE;
STEP01: jj=0; goto STEP02;
STEP03:if ( jj < 0 ) goto STEP04;kk=0; goto STEP02;
STEP04:if ( carry == 0 ) goto STEP05;k='1';fputc(k,fpG);
STEP05:fclose(fpE);fclose(fpC);fclose(fpG);return 0;}

int main(void){EPCG();return 0;}
```

**File E (E.txt) の Data に File C (C.txt)
の Data を加算して、File G (G.txt) に
保存する。**

G = E + C

(右上位表示)

要素Program (Function)

- Function01(AONE).txt
- Function02(DONE).txt
- Function03(EZERO).txt
- Function04(A10C).txt
- Function05(D100C).txt
- Function06(CTOA).txt
- Function07(CTOD).txt
- Function08(CTOE).txt
- Function09(CTOF).txt
- Function10(DTOE).txt
- Function11(FTOD).txt
- Function12(GTOE).txt
- Function13(BPXC).txt
- Function14(CPXA).txt
- Function15(CPEB).txt
- Function16(EPCG).txt
- Function17(CXE).txt
- Function18(DSEC).txt
- Function19(GETNEXTX)...
- Function20(Main).txt





















Function17(CXE):

File C (C.txt) の Data を
x 倍して File E (E.txt) に 保存する。

$$E = x * C \quad (\text{右上位表示})$$

```
int CXE(void)  {int ix;
                if( x > 0 ) goto STEP1;EZERO(); return 0;
STEP1:         CTOE();if( x == 1 ) return 0;ix=2;
MOREX:        EPCG(); GTOE(); if (ix == x ) return 0;
                ix=ix+1; goto MOREX;}
int main(void){CXE();return 0;}
```

要素Program (Function)

-  Function01(AONE).txt
-  Function02(DONE).txt
-  Function03(EZERO).txt
-  Function04(A10C).txt
-  Function05(D100C).txt
-  Function06(CTOA).txt
-  Function07(CTOD).txt
-  Function08(CTOE).txt
-  Function09(CTOF).txt
-  Function10(DTOE).txt
-  Function11(FTOD).txt
-  Function12(GTOE).txt
-  Function13(BPXC).txt
-  Function14(CPXA).txt
-  Function15(CPEB).txt
-  Function16(EPCG).txt
-  Function17(CXE).txt
-  Function18(DSEC).txt
-  Function19(GETNEXTX)...
-  Function20(Main).txt

Function18(DSEC):

File D(D.txt) の Dataから
File E(E.txt) の Dataを
減算して、File C(C.txt)
に保存する。

$$C = D - E$$

(右上位表示)

もし $D < E$ なら Cの数字
の頭に補数符号 '/' を添付。

$D \geq E$ なら DESC=1,
 $D < E$ なら DESC=-1 を返す。

これは 10進法の 補数減算法である:

例:

	73	
-)	195	

	/878 = 1000 - 878 = 122	
	195	878
-)	73	+) 122
	-----	-----
	122	1000

```
#include <stdio.h>

int DSEC(void){char j,k;int carry,jj,kk;int t=1;
                FILE *fpD;FILE *fpE;FILE *fpC;
                fpD=fopen("D.txt","r");fpE=fopen("E.txt","r");
                fpC=fopen("C.txt","w"); carry=0;

ONCEMORE: j=fgetc(fpD); k=fgetc(fpE);kk=-1;jj=-1;

if ( j == '1' ) jj=1;if ( j == '2' ) jj=2;if ( j == '3' ) jj=3;
if ( j == '4' ) jj=4;if ( j == '5' ) jj=5;if ( j == '6' ) jj=6;
if ( j == '7' ) jj=7;if ( j == '8' ) jj=8;if ( j == '9' ) jj=9;
if ( j == '0' ) jj=0;

if ( k == '1' ) kk=1;if ( k == '2' ) kk=2;if ( k == '3' ) kk=3;
if ( k == '4' ) kk=4;if ( k == '5' ) kk=5;if ( k == '6' ) kk=6;
if ( k == '7' ) kk=7;if ( k == '8' ) kk=8;if ( k == '9' ) kk=9;
if ( k == '0' ) kk=0;

if ( kk < 0 ) goto STEP03;if ( jj < 0 ) goto STEP01;

STEP02:kk=jj-kk-carry;carry=0;if ( kk < 0 ) { carry = 1 ; kk=10+kk; }

if ( kk==0 ) k='0';if ( kk==1 ) k='1';if ( kk==2 ) k='2';
if ( kk==3 ) k='3';if ( kk==4 ) k='4';if ( kk==5 ) k='5';
if ( kk==6 ) k='6';if ( kk==7 ) k='7';if ( kk==8 ) k='8';
if ( kk==9 ) k='9';

fputc(k,fpC);goto ONCEMORE;

STEP01: jj=0; goto STEP02;
STEP03:if ( jj < 0 ) goto STEP04;kk=0; goto STEP02;
STEP04:if ( carry == 0 ) goto STEP05;t=-1;k='/';fputc(k,fpC);
STEP05:fclose(fpD);fclose(fpE);fclose(fpC);return t;}

int main(void){int t;t=DSEC();printf("t=%d\n",t); getchar();return 0;}
```

$$(\sqrt{2})^2 - (\text{近似値})^2 = (\text{2乗誤差})$$

$$D(i) = 100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} > 0$$

Function19(GETNEXTX):

という便利な関係に注目して
数値計算 Algorizm を構築する。

最終的に Program02~19
をすべて使って次の桁 x
を このProgram20で求める。

```
int GETNEXTX(void){char s;int ix,t;x=0;
D100C();CTOD(); /* D=100*D */
A10C();CTOE(); CPEB(); /* B= 20*A */
NEXTTRY:
    BPXC(); /* C= 20*A+x */
    CXE(); /* E= x*(20*A+x) */
    t=DSEC(); /* C= D - E */
if (t < 0 ) goto XEND;
CTOF();x=x+1;goto NEXTTRY; /* F= D - E */
XEND: x=x-1; FTOD(); A10C(); CPXA();return 0;}

int main(void){char s;int i=1;
MOREX:GETNEXTX();printf(" %d %d ",i,x);i=i+1;
s=getchar(); if( s == 's' )return 0;goto MOREX;}
```

$$(\sqrt{2})^2 - (\text{近似値})^2 = (\text{2乗誤差})$$

$$D(i) = 100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} > 0$$

Function20(MAIN):

という便利な関係に注目して
数値計算 Algorizm を構築する。

最終的に Function01~19
をすべて使って次の桁 x
をこのFunction20(Main)
で求める。

MAIN PROGRAM は
入出力の形式を定義して
出力が見やすいように
している。

最終 Output は R.txt に入る。

全体 NUM01(ROOT2)。

```
int main(void){char j;int istop ;int NJ;int m10;

FILE *fpR;fpR=fopen("R.txt","w");j='1';fputc(j,fpR);j='.';fputc(j,fpR);
AONE();DONE();
ONCEMOREJ:j=getchar();NJ = 1;

if ( j == '1' ) NJ =9;if ( j == '2' ) NJ =99;if ( j == '3' ) NJ =999;
if ( j == '4' ) NJ = 9999;if ( j == '5' ) NJ = 99999;
if ( j == 's' ) { fclose(fpR);return 0;} istop=m+NJ;

ONCEMORE:if ( m == istop ) goto ONCEMOREJ;

m=m+1;GETNEXTX();

if ( x ==0 ) j='0';if ( x ==1 ) j='1';if ( x==2 ) j='2';
if ( x ==3 ) j='3';if ( x ==4 ) j='4';if ( x==5 ) j='5';
if ( x ==6 ) j='6';if ( x ==7 ) j='7';if ( x==8 ) j='8';
if ( x ==9 ) j='9';fputc(j,fpR);

m10=m-(m/10)*10; if( m10==0 ) fputc(' ',fpR) ;
m10=m-(m/1000)*1000; if( m10==0 ) fputc('\n',fpR) ;
m10=m-(m/100)*100; if( m10==0 ) {fputc('\n',fpR) ;
fputc(' ',fpR) ;fputc(' ',fpR) ;}

if (m>9 ) goto II;printf("00000%d %d \n",m,x);goto ONCEMORE;

II:if ( m>99 ) goto III;printf("0000%d %d \n",m,x);goto ONCEMORE;
III:if ( m>999 ) goto IIII;printf("000%d %d \n",m,x);goto ONCEMORE;
IIII:if ( m>9999 ) goto IIIII;printf("00%d %d \n",m,x);goto ONCEMORE;
IIIII:if ( m>99999 ) goto IIIIII;printf("0%d %d \n",m,x);goto ONCEMORE;
IIIIII:if ( m>999999 ) return 0;printf("%d %d \n",m,x);goto ONCEMORE;}
```

$$(\sqrt{2})^2 - (\text{近似値})^2 = (\text{2乗誤差})$$

$$D(i) = 100 * D(i-1) - X(i) \{20 * A(i-1) + X(i)\} > 0$$

という便利な関係に注目して
数値計算 Algorizm を構築する。

最終的に Function01~19 をすべて使っ
て次の桁 x を このFunction20(Main)で
求める。

MAIN PROGRAM は 入出力の形式を定
義して出力が見やすいようにしている。

最終 Output は R.txt に入る。

全体 Program は NUM02(ROOT2) 。

/*****(Output Example of R.txt)*****/

```
1.4142135623 7309504880 1688724209 6980785696 7187537694 8073176679 7379907324 7846210703 8850387534 3276415727
3501384623 0912297024 9248360558 5073721264 4121497099 9358314132 2266592750 5592755799 9505011527 8206057147
0109559971 6059702745 3459686201 4728517418 6408891986 0955232923 0484308714 3214508397 6260362799 5251407989
6872533965 4633180882 9640620615 2583523950 5474575028 7759961729 8355752203 3753185701 1354374603 4084988471
6038689997 0699004815 0305440277 9031645424 7823068492 9369186215 8057846311 1596668713 0130156185 6898723723
5288509264 8612494977 1542183342 0428568606 0146824720 7714358548 7415565706 9677653720 2264854470 1585880162
0758474922 6572260020 8558446652 1458398893 9443709265 9180031138 8246468157 0826301005 9485870400 3186480342
1948972782 9064104507 2636881313 7398552561 1732204024 5091227700 2269411275 7362728049 5738108967 5040183698
6836845072 5799364729 0607629969 4138047565 4823728997 1803268024 7442062926 9124859052 1810044598 4215059112
0249441341 7285314781 0580360337 1077309182 8693147101 7111168391 6581726889 4197587165 8215212822 9518488472
```

小数点以下1000けたの $\sqrt{2}$ の値

1.4142135623 7309504880 1688724209 6980785696 7187537694 8073176679 7379907324 7846210703 8850387534 3276415727
3501384623 0912297024 9248360558 5073721264 4121497099 9358314132 2266592750 5592755799 9505011527 8206057147
0109559971 6059702745 3459686201 4728517418 6408891986 0955232923 0484308714 3214508397 6260362799 5251407989
6872533965 4633180882 9640620615 2583523950 5474575028 7759961729 8355752203 3753185701 1354374603 4084988471
6038689997 0699004815 0305440277 9031645424 7823068492 9369186215 8057846311 1596668713 0130156185 6898723723
5288509264 8612494977 1542183342 0428568606 0146824720 7714358548 7415565706 9677653720 2264854470 1585880162
0758474922 6572260020 8558446652 1458398893 9443709265 9180031138 8246468157 0826301005 9485870400 3186480342
1948972782 9064104507 2636881313 7398552561 1732204024 5091227700 2269411275 7362728049 5738108967 5040183698
6836845072 5799364729 0607629969 4138047565 4823728997 1803268024 7442062926 9124859052 1810044598 4215059112
0249441341 7285314781 0580360337 1077309182 8693147101 7111168391 6581726889 4197587165 8215212822 9518488472

10進法小数で $\sqrt{2}$ を計算する flow chart

