第5章　プログラミングの基礎

5.1　アルゴリズム

ある問題を解決するための手段を**アルゴリズム**(algorithm) といいます。実は、すでに紹介した10 進数を r 進数に変換した手順やシフト演算と足し算のみで掛け算や割り算を実行する手順もアルゴリズムです。以下の 2 つの問題のアルゴリズムを考えてみましょう。

最初の問題として、正の整数の平方根を求めるアルゴリズムを考えてみましょう。奇数の数列の和は

$$\sum_{i=1}^{n}(2i-1) = 1+3+\cdots+(2n-1) = n^2$$

で与えられ、このことを利用すると正の整数 N の平方根を求めることができます。例として、$N=5$ とすると

$$5 = (1+3)+1 = \{1+(2\cdot 2-1)\}+1$$

となるので、平方根の整数部分が 2 となります。答えがもう一桁必要な場合は、5 を 100 倍して

$$\begin{aligned} 500 &= 20^2+100 = (1+3+\cdots+39)+41+43+16 \\ &= \{(1+3+\cdots+39+41+(2\cdot 22-1)\}+16 \end{aligned}$$

となり、22 を 10 で割ると 2.2 となります。さらにもう一桁必要な場合は、5 を 10000 倍して

$$\begin{aligned} 50000 &= 220^2+1600 = (1+3+\cdots+439)+441+443+445+271 \\ &= \{(1+3+\cdots+441+443+(2\cdot 223-1)\}+271 \end{aligned}$$

となり、223 を 100 で割ると 2.23 となります。この操作を繰り返せば 5 の平方根を必要な桁数まで求めることができます。

2 つ目の問題として、正の整数 x, y の最大公約数 (greatest common divisor) を求めるアルゴリズムを考えてみましょう。ただし、$x > y$ とします。初等的な代数を学習したみなさんであれば、**ユークリッドの互除法**という有名なアルゴリズムがあることをよく知っていて、

$$\gcd(x,y) = \begin{cases} x & ,y = 0 \text{ のとき} \\ \gcd(y, x \bmod y) & ,y \neq 0 \text{ のとき} \end{cases}$$

で与えられ、$y = 0$ になるまで繰り返し gcd を計算すれば最大公約数が求まります。ただし、$x \bmod y$ は x を y で割った余りを表します。例として、1234 と 56 の最大公約数を求めてみましょう。上の関係式より、

$$\gcd(1234, 56) = \gcd(56, 2) = \gcd(2, 0)$$

従って、123 と 56 の最大公約数は 2 になります。

問題 1　7 の平方根を上記のアルゴリズムを使って求めなさい。

問題 2　3235 と 875 の最大公約数をユークリッドの互助法を用いて計算しなさい。

問題 3　正整数の 3 乗根を求めるアルゴリズムを考えなさい。

問題 4　最小公倍数を求めるアルゴリズムを考えなさい。

5.2　流れ図

流れ図(flow chart) はアルゴリズムを図式化したもので、データの流れ・判定条件・実行の推移などを表現します。流れ図は記号を線で繋いだもので、記号は JIS 規格で決まっており、主なものを表に挙げておきます。左上から下に順に、端子・接合子・入力・準備・処理・定義済み処理・ループ端上・ループ端下・判断を表します。

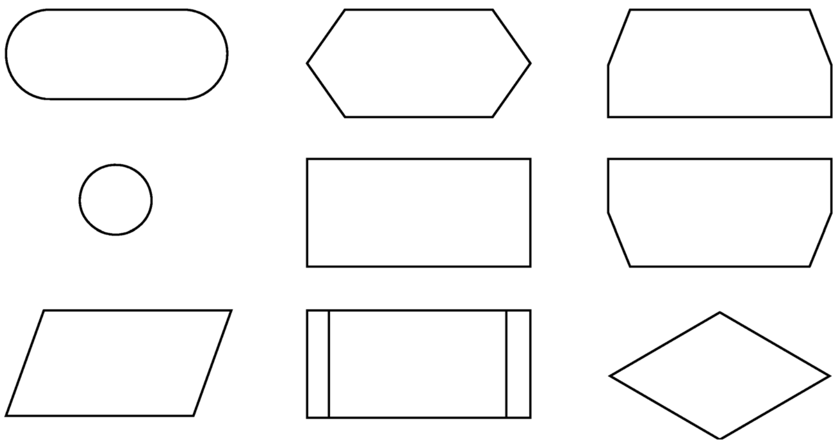

図 5.1: 流れ図記号

前節のアルゴリズムを流れ図にすると次のようになります。

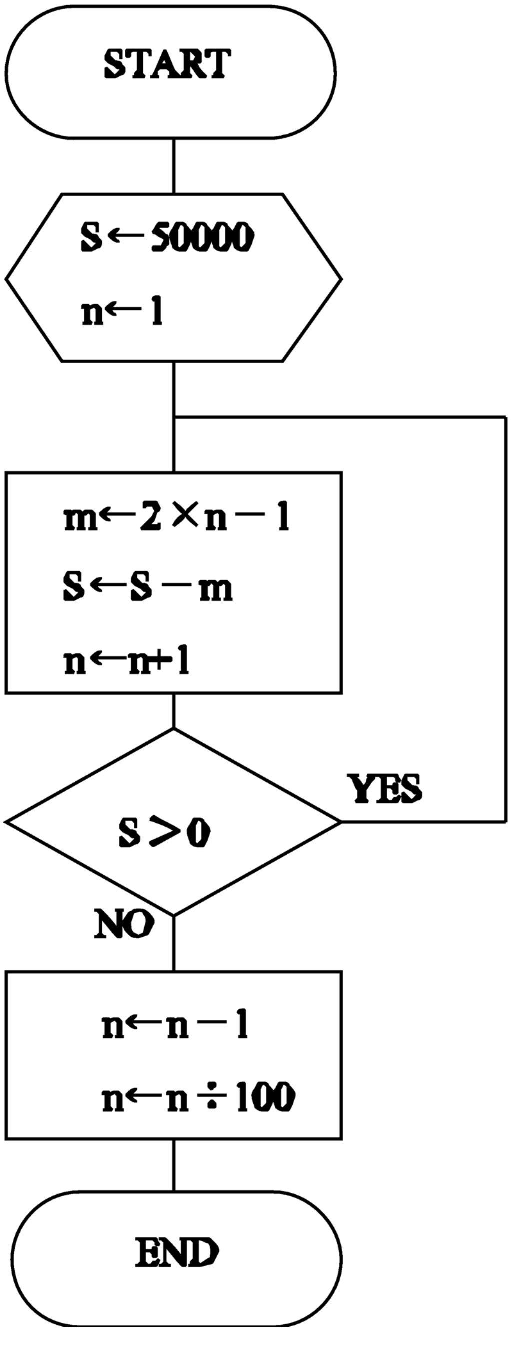

図 5.2: 5 の平方根 2.23 を求める流れ図

矢印は代入を表します。

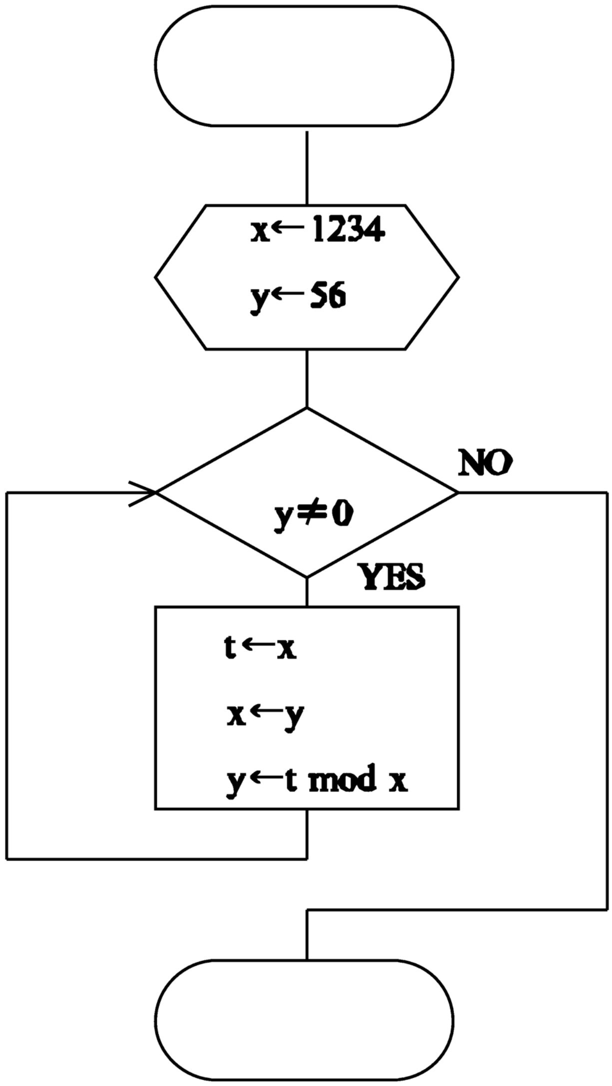

図 5.3: 正の整数 x, y の最大公約数を求める流れ図

問題 1　1 から 50 までの和を求める流れ図を作成しなさい。

問題 2　最小公倍数を求める流れ図を作成しなさい。