

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。

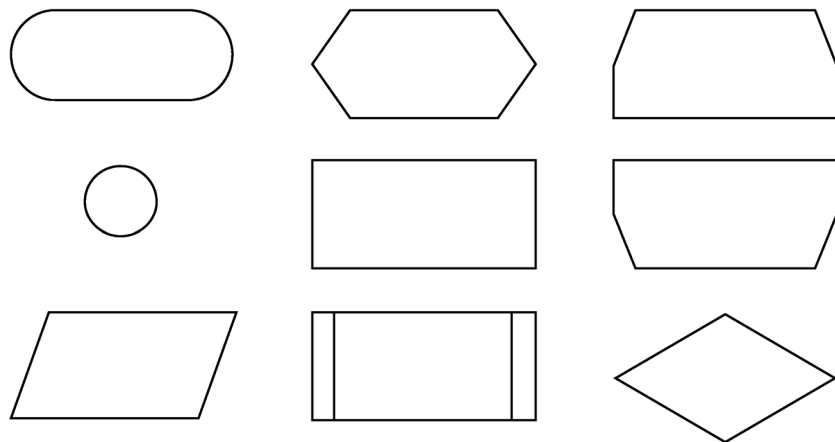


図 5.1: 流れ図記号

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。

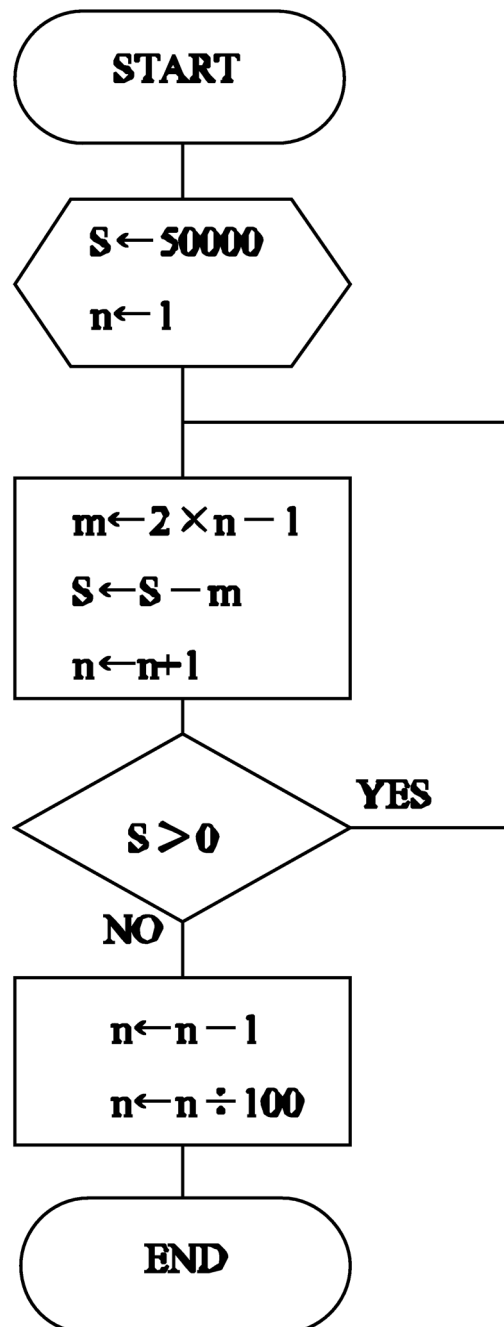


図 5.2: 5 の平方根 2.23 を求める流れ図

矢印は代入を表します。

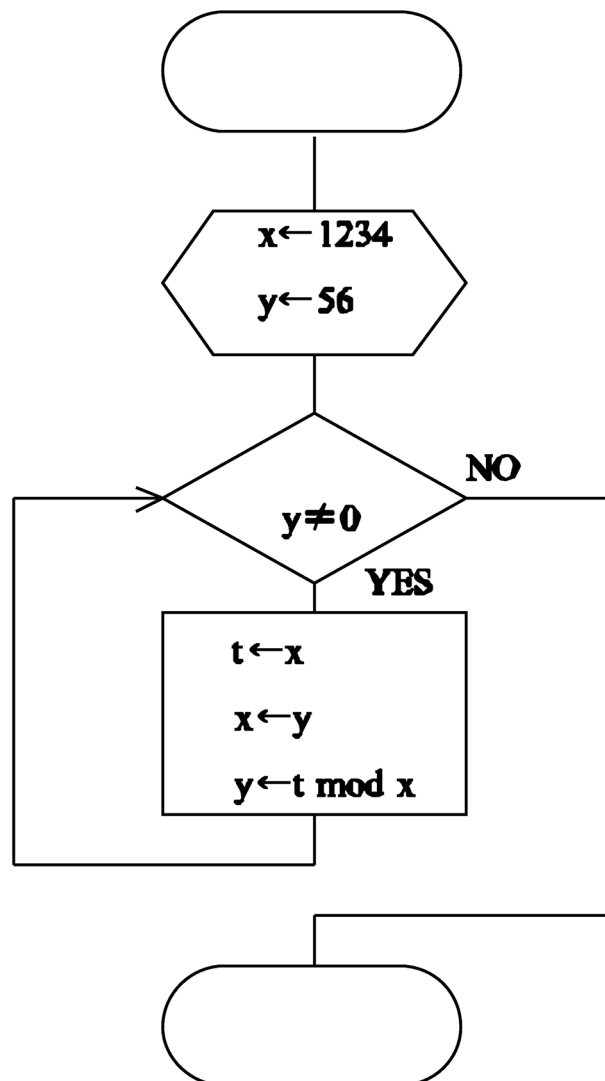


図 5.3: 正の整数  $x, y$  の最大公約数を求める流れ図

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。