



# 1. フローチャートとトレース

## 1.1. フローチャート

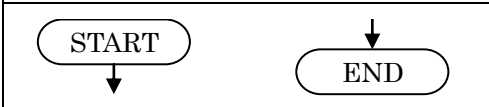
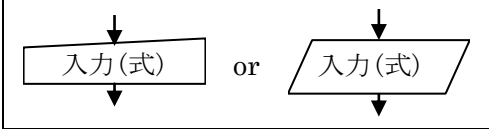
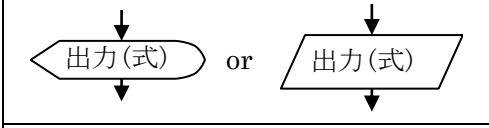
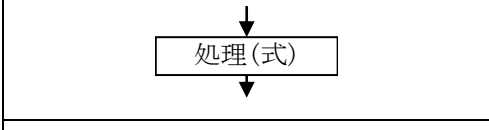
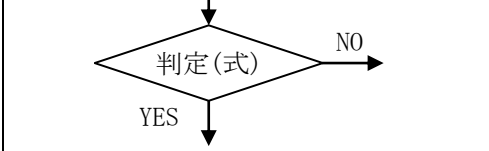
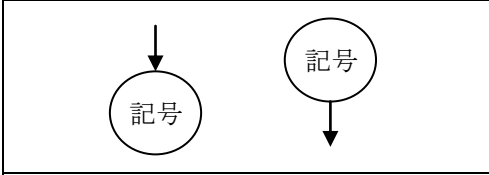
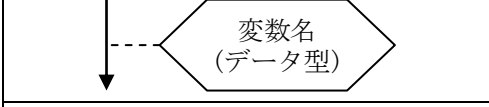
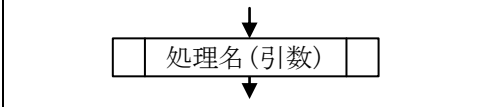
フローチャート：アルゴリズムを図形や矢印を用いて視覚的に表現した図式

処理の種類や内容

処理の流れ：原則は、  原則は、上から下へ、左から右へ

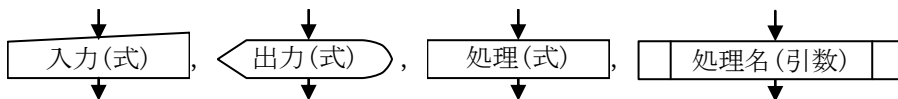
原則的な流れの向きでは、線でも良い。

### ● フローチャートの用いる図形とそれが表わす意味

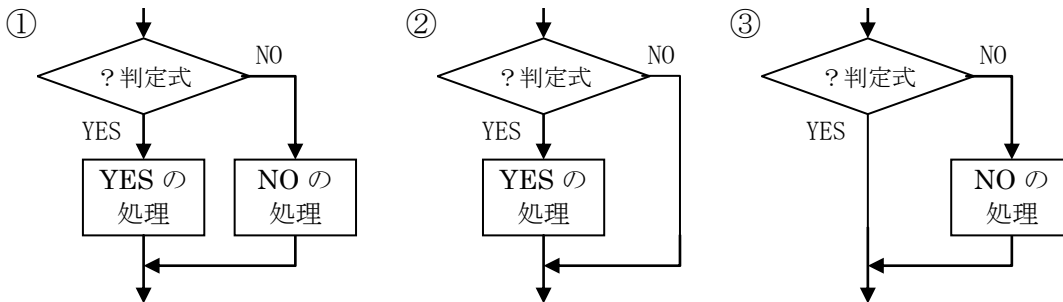
記号	名称	処理内容または意味
	端子	フローチャートの開始と終了
	入力記号	データの入出力
	出力記号	データの出力、表示
	処理記号	計算、一般の処理
	判定記号	選択処理や反復処理の判定 選択処理のときは「?」を、反復処理にときは「@」を付けることとする(ローカルルール).
	結合子	フローチャートの分離・連結 同じ記号を持つ結合子で連結されていると考える.
	宣言記号	変数、配列等の宣言 (省略されることも多い)
	外部定義 処理記号	定義済み外部処理の呼び出し

● フローチャートでの処理構造

☆ 順次処理



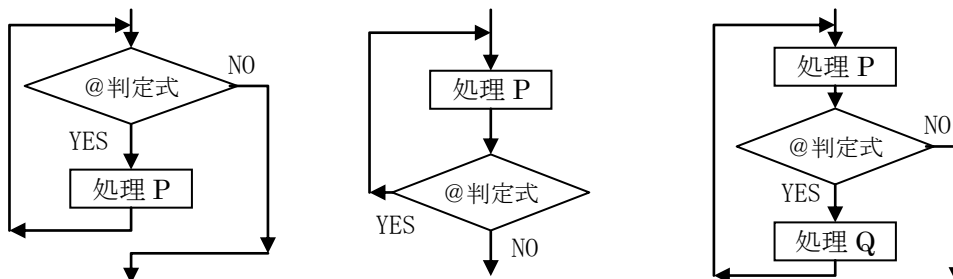
☆ 選択処理：判定式に、『?』をつけることによる。



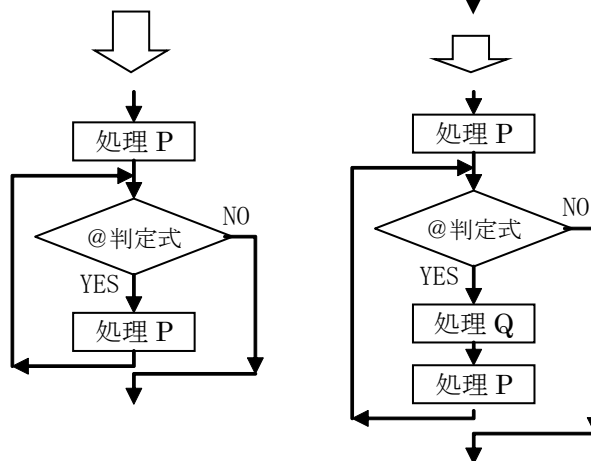
(注) パターン③は判定式を否定すれば、②のパターンに変換できる。

☆ 反復処理：判定式に、『@』をつけることによる。

①：前判定反復処理      ②：後判定反復処理      ③：中判定反復処理



(注) パターン②、③の処理は、  
①の前判定パターンに変換できる。

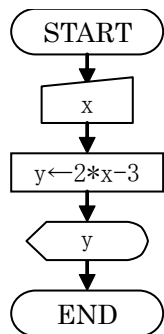


(1.1) 例題

以下の各処理をフローチャートで描く。

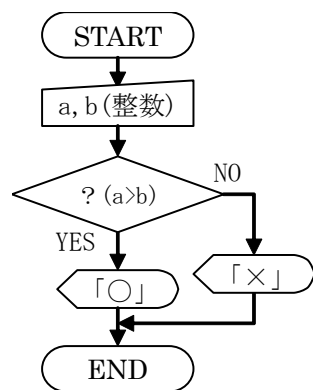
- (1) 変数  $x$  に入力された数に対して、 $2x-3$  を計算して変数  $y$  に代入し、 $y$  の値を出力する。

- |   |
|---|
| ① 変数 $x$ に値を入力。(入力： $x$ )<br>② 変数 $y$ に $2x-3$ を代入。( $y \leftarrow 2x-3$ )<br>③ 変数 $y$ の値を出力。(出力： $y$ ) |
|---|



- (2) 2つの変数  $a, b$  にそれぞれ整数値を入力し、 $a > b$  のとき「○」を、そうでないとき「×」を出力する。

- |   |
|---|
| ① 入力： $a, b$ (整数)<br>② $a > b$ を判定<br>②YES 出力：「○」<br>②NO 出力：「×」 |
|---|



- (3) 1 から 10 までの各整数の 2 乗を計算し出力する。

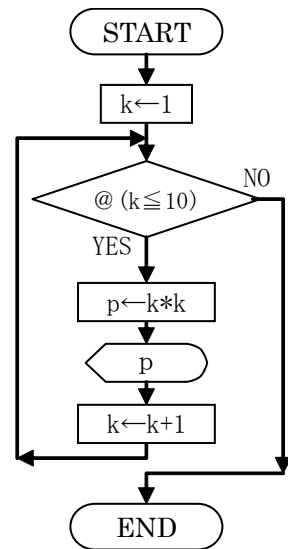
- |   |
|---|
| ① $k \leftarrow 1$ ( $k$ は繰り返し回数に対応)<br>② $k \leq 10$ の間、以下の処理を繰り返す。<br>②1 $p \leftarrow k * k$<br>②2 出力： $p$<br>②3 $k \leftarrow k + 1$<br>(次の繰り返し処理に備えて、 $k$ の更新) |
|---|

(注) 『 $k$  の更新』

$k$  の値を新しくする事。

今の場合、 $k$  の値を 1 ずつ増やしながら処理を繰り返していくので、 $k+1$  を改めて  $k$  に代入することが、 $k$  の更新になる。

ちなみに、 $k$  の値を 1 ずつ減らすときは、 $k$  の更新式は『 $k \leftarrow k - 1$ 』となる。

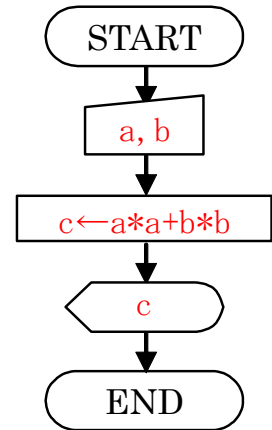


1.2. フローチャート練習

以下の設問の各処理のフローチャートの空欄を埋めよ.

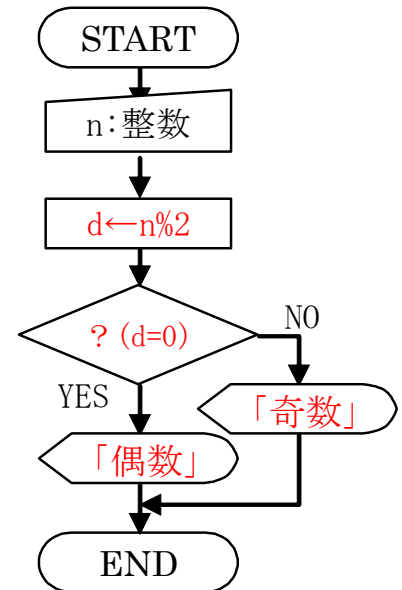
- (1) 2つの変数 a, b に数を入力し, それぞれの値の 2 乗の和を変数 c に代入( $c \leftarrow a*a + b*b$ )し, その値を出力する.

- |                            |
|----------------------------|
| ① 入力 : a, b                |
| ② $c \leftarrow a*a + b*b$ |
| ③ 出力 : c                   |



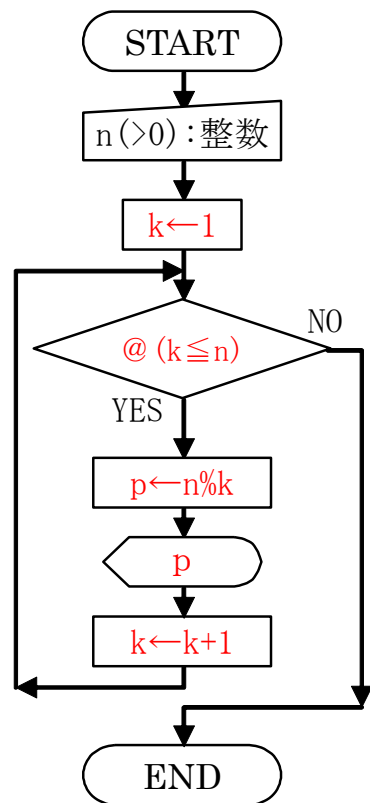
- (2) 変数 n に整数値を入力して, それを 2 で割った余りを変数 d に代入する. そのとき,  $d=0$  ならば「偶数」を, そうでないならば「奇数」を出力する.

- |  |
|--|
| ① 入力(整数) : n   |
| ② $d \leftarrow n \% 2$ ( $a \% b \dots a \div b$ の余りを求める演算記号) |
| ③ $d=0$ の判定  |
| ③ YES 出力 : 「偶数」  |
| ③ NO 出力 : 「奇数」   |



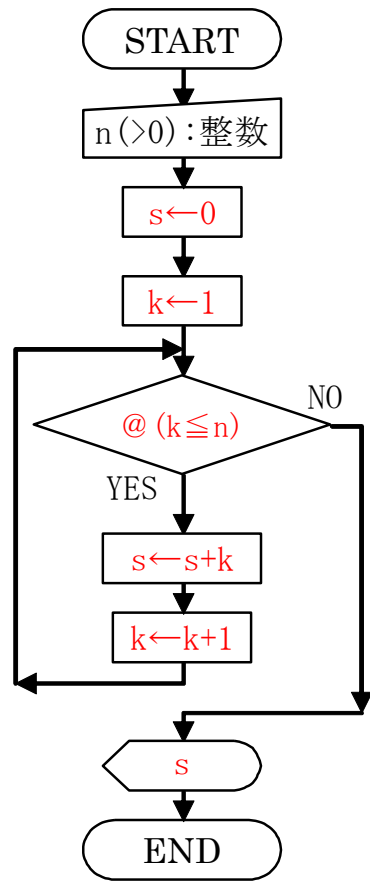
(3) 変数  $n$  に 0 より大きい整数値を入力して、変数  $k$  を 1 から  $n$  まで順に動かすとき、各  $k$  に対して  $n \div k$  の余りを計算して出力する。

- ① 入力 (整数 $>0$ ) :  $n$
- ②  $k \leftarrow 1$
- ③  $k \leq n$  の間、以下の処理を繰り返す。
  - ③1  $p \leftarrow n \% k$
  - ③2 出力 :  $p$
  - ③3  $k \leftarrow k + 1$



(4) 変数  $n$  に 0 より大きい整数値を入力して、1 から  $n$  までの総和を求め、その値を出力する。

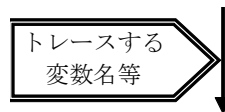
- ① 入力 (整数 $>0$ ) :  $n$
- ②  $s \leftarrow 0$
- ③  $k \leftarrow 1$
- ④  $k \leq n$  の間、以下の処理を繰り返す。
  - ④1  $s \leftarrow s + k$
  - ④2  $k \leftarrow k + 1$
- ⑤ 出力 :  $s$



1.3. トレース

トレース：処理の流れに沿って、変数の値や出力の変化を調べること。

トレース点：処理の流れやフローチャート内において、トレースする箇所のこと。右のように表わす。



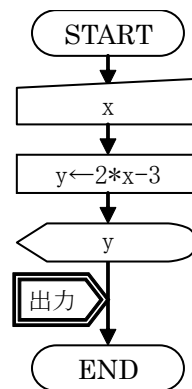
トレース表：処理の流れやフローチャート内のトレース点における変数等の値を表したもの。

(1.2) 例題

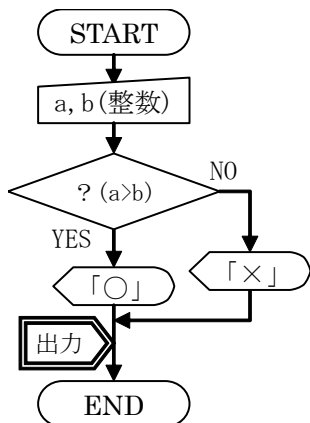
例題(1.1)の各フローチャートのトレース点における変数値のトレース表を作成する。

(1) 変数 x に入力された数に対して、 $2x-3$  を計算して変数 y に代入し、y の値を出力する。

入力	$x \leftarrow 3$	$x \leftarrow 0$	$x \leftarrow -5$	$x \leftarrow -2.5$
出力	<b>3</b>	<b>-3</b>	<b>-13</b>	<b>2</b>

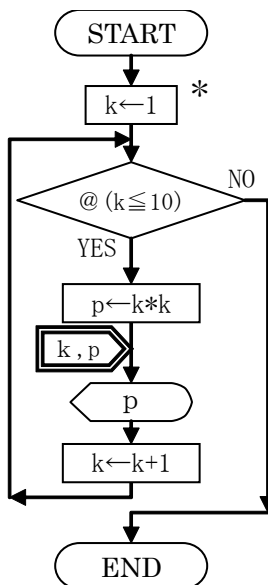


(2) 2 つの変数 a, b にそれぞれ数を入力し、 $a > b$  のとき「○」を、そうでないとき「×」を出力する。



入力	$a \leftarrow 2, b \leftarrow 5$	$a \leftarrow 3, b \leftarrow -1$
出力	<b>×</b>	<b>○</b>

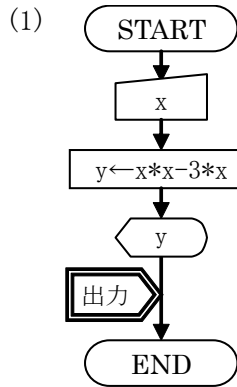
(3) 1 から 10 までの各整数の 2 乗を計算し出力する。



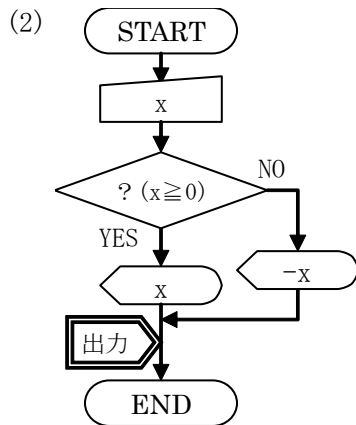
*の直後	1	
通過回数	k	p
1	<b>1</b>	<b>1</b>
2	<b>2</b>	<b>4</b>
3	<b>3</b>	<b>9</b>
4	<b>4</b>	<b>16</b>
5	<b>5</b>	<b>25</b>
6	<b>6</b>	<b>36</b>
7	<b>7</b>	<b>49</b>
8	<b>8</b>	<b>64</b>
9	<b>9</b>	<b>81</b>
10	<b>10</b>	<b>100</b>

1.4. トレース練習

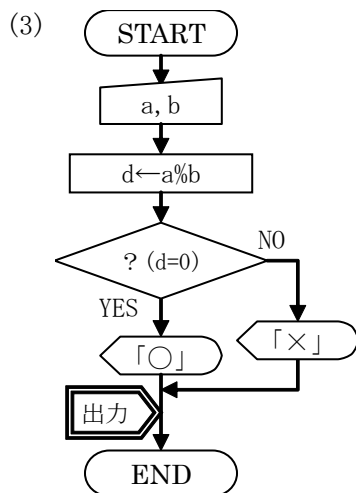
以下の設問のフローチャートのトレース点におけるトレース表を完成させよ.



入力	$x \leftarrow 1$	$x \leftarrow -1$	$x \leftarrow 0$	$x \leftarrow -3$
出力	-2	4	0	18

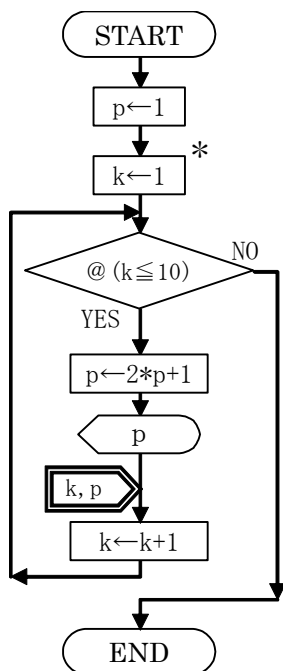


入力	$x \leftarrow 3$	$x \leftarrow -2.5$	$x \leftarrow 0$	$x \leftarrow -7$
出力	3	2.5	0	7



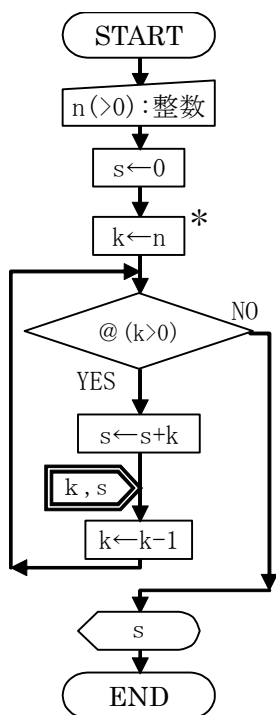
入力	$a \leftarrow 10, b \leftarrow 3$	$a \leftarrow 8, b \leftarrow 4$	$a \leftarrow 5, b \leftarrow 20$
出力	×	○	×

(4)



*の直後	1	1
通過回数	k	p
1	1	3
2	2	7
3	3	15
4	4	31
5	5	63
6	6	127
7	7	255
8	8	511
9	9	1023
10	10	2047

(5)



入力	n ← 10		入力	n ← 6	
*の直後	10	0	*の直後	6	0
通過回数	k	s	通過回数	k	s
1	10	10	1	6	6
2	9	19	2	5	11
3	8	27	3	4	15
4	7	34	4	3	18
5	6	40	5	2	20
6	5	45	6	1	21
7	4	49	7		
8	3	52	8		
9	2	54	9		
10	1	55	10		