

図記号

① (端子)



プログラムの開始/終了やスタート/ゴールを表す。一般的に”端子”と呼ばれフローの(**始まりと終わり**)に配置する必要がある

② (処理)



処理、作業、手続きを表す際に配置。基本的には1ボックスにつき、(**1つの処理内容を記載**)

③ (判断)



「(Yes/No)」が答えとなる判断や、複数の選択肢に分かれる判断を表す記号。ひし形の図形で”判断”もしくは”条件分岐”と呼ばれ、2つ以上の判断処理を表す際に配置。

④ (手入力)



キーボードや(**デバイス**)経由のデータの手動入力

⑥ (表示)



ディスプレイへの表示を表す

⑤ (入出力)



媒体を指定しないデータやファイルなどの参照や書込み、入出力を行う機能を表す。”データ”もしくは”入出力”、”(I/O)”とも呼ばれる。主にデータベースとファイルを分けて図示したい際に配置。

⑦ (書類)



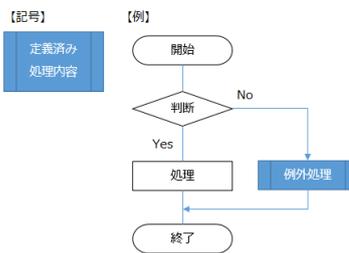
書類の入出力を表す。書類の入力例：レポート、メール、発注書の受領など
出力例：プレゼンテーション、メモ、通信文の作成

⑧ (ループ)



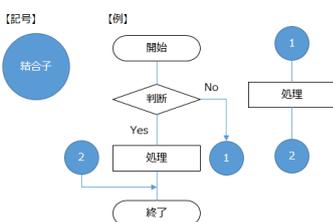
何度も繰り返して行う処理を表す。台形のような六角形の図形で、繰り返し条件と繰り返し終了条件を表す際に配置。ボックスの中に繰り返し名と条件を記入し、間に(**処理**)を挟む。

⑨ (定義済み処理)



処理の一部を別フローチャートと分けて作成する場合に配置する。フローチャートを1枚にまとめる際に長くなるケースは、この記号で一部フローを分けて記載することが多い。

⑩ (結合子)

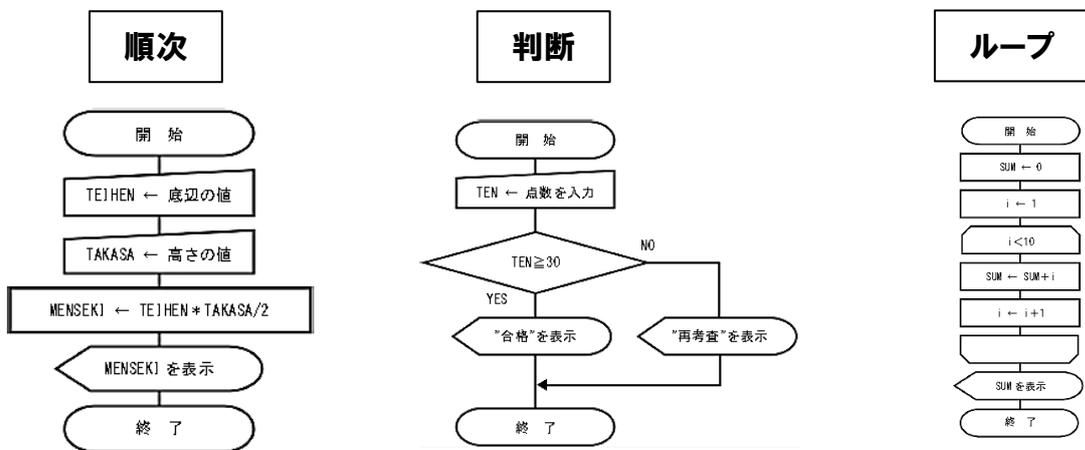


フローチャートが長くなったり、複雑になったりした際に同じフローチャート内で(**別の処理**)に飛ばす場合に配置する。結合子の中に参照先を記載することが多い。

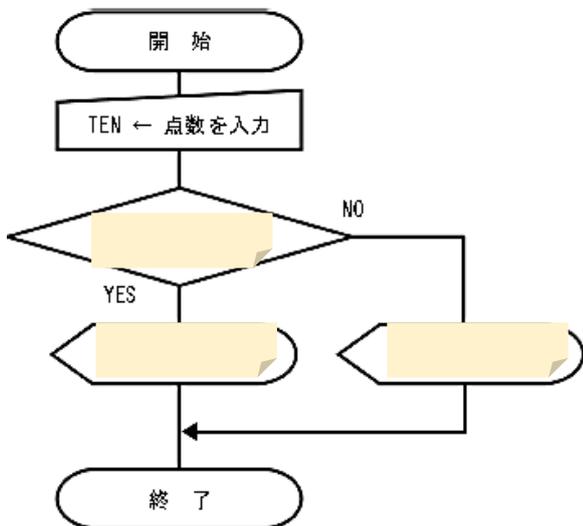
図記号クイズ!

	定義済み処理		結合子
	ループ		端子
	判断		書類
	処理		入出力
			表示
			手入力

(1) 基本的な流れ (2) 判断を使った流れ (3) ループを使った流れ



問題. 図 3-76 の流れ図を使用して、
25 点以下なら“不合格”、そうでない
場合は“合格”を表示するように書き
換えましょう。

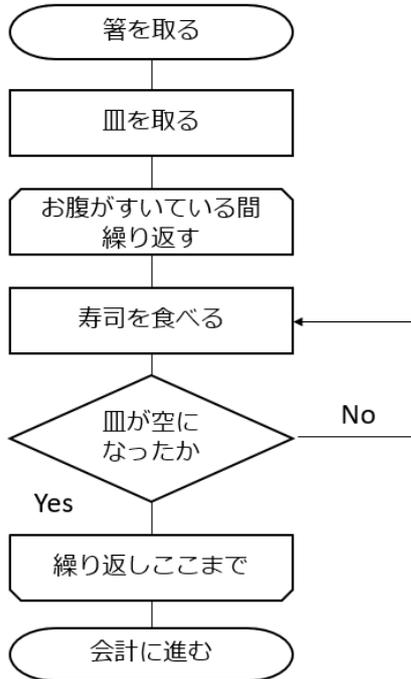


問題. 朝起きて家を出るまでを流れ図で描きま
 しょう。

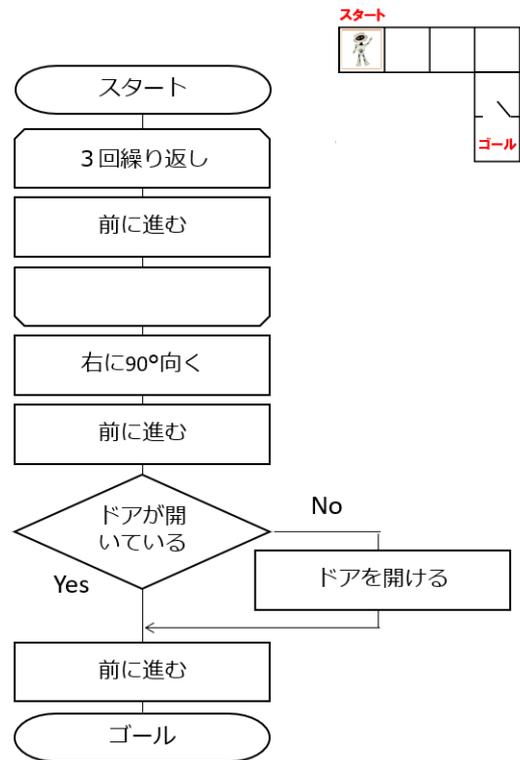
- ① 入口(開始)は1つ
- ② 出口(終了)は1つ
- ③ 線は上から下へ書く
- ④ なるべく判断をいれる
- ⑤ 分岐するときは右に行く

問題. お腹がすいている間、食べ放題の回転寿司を食べ続ける流れを図にしましょう。

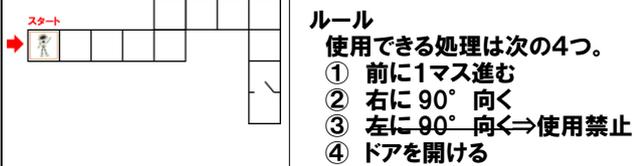
- ヒント
- ① 繰り返す作業は何だろう？
 - ② 判断すべきポイントはどこだろう？



問題. 下図のような通路がある。次の4つのルールに従って、ゴールに到達するまで流れ図を描きましょう。

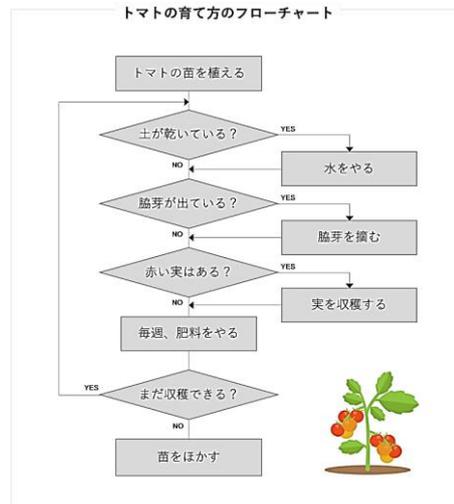


発展問題. 下図のような通路がある。次の3つのルールに従って、ゴールに到達するまでの流れ図を描きましょう。



問題. 身近な出来事を題材に簡単な流れ図を描きましょう。

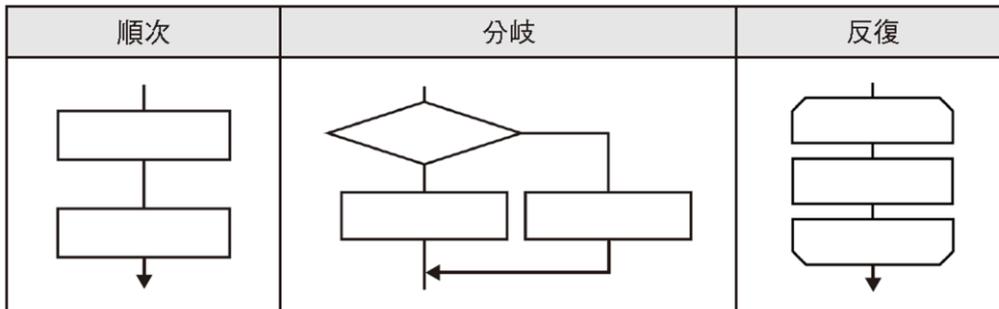
- ルール
- ① 入口(開始)は1つ
 - ② 出口(終了)は1つ
 - ③ 線は上から下へ書く
 - ④ なるべく判断をいれる
 - ⑤ 分岐するときは右に行く
 - ⑥ 題材は何でも OK !



(1) アルゴリズムとは？

(問題を解決) するための (手順) や (方法)

(2) 構造化プログラミング

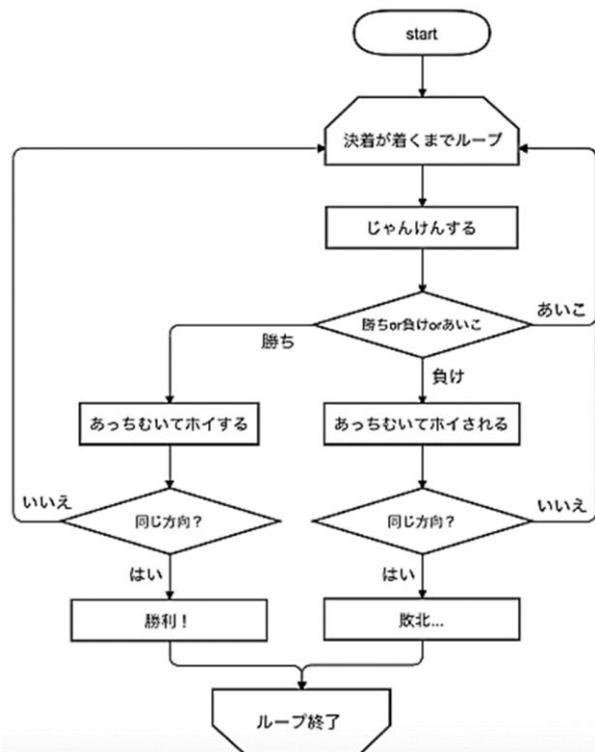


発展問題. 「順次」「分岐」「反復」の構造をすべて使用して、身近な出来事を題材に簡単な流れ図を作成しましょう。

- ① 入口(開始)は1つ
- ② 出口(終了)は1つ
- ③ 線は上から下へ書く
- ④ 順次、分岐、反復の構造を入れる
- ⑤ 分岐するときは右に行く
- ⑥ 題材は何でもOK!
- ⑦ 早くできた人は「定義済み処理」を加えて作成してみよう!

問題. 「あっち向いてホイ」の遊びを流れ図で描きましょう。

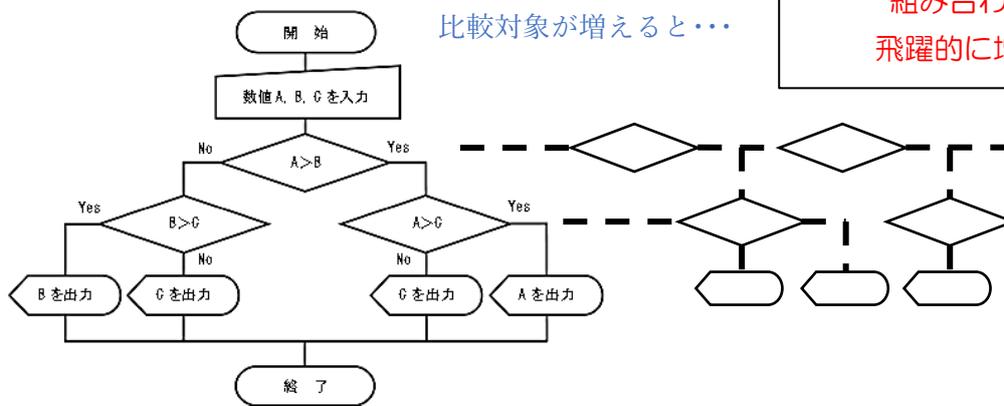
- ヒント
1. じゃんけんする
 2. 勝つ or 負ける or あいこ
 - a. [勝つ] あっちむいてホイする
 - i. 同じ方向?
 1. [はい] 勝利!
 2. [いいえ] もう一度じゃんけん
 - b. [負ける] あっちむいてホイされる
 - i. 同じ方向?
 1. [はい] 敗北...
 2. [いいえ] もう一度じゃんけん
 - c. [あいこ] もう一度じゃんけん



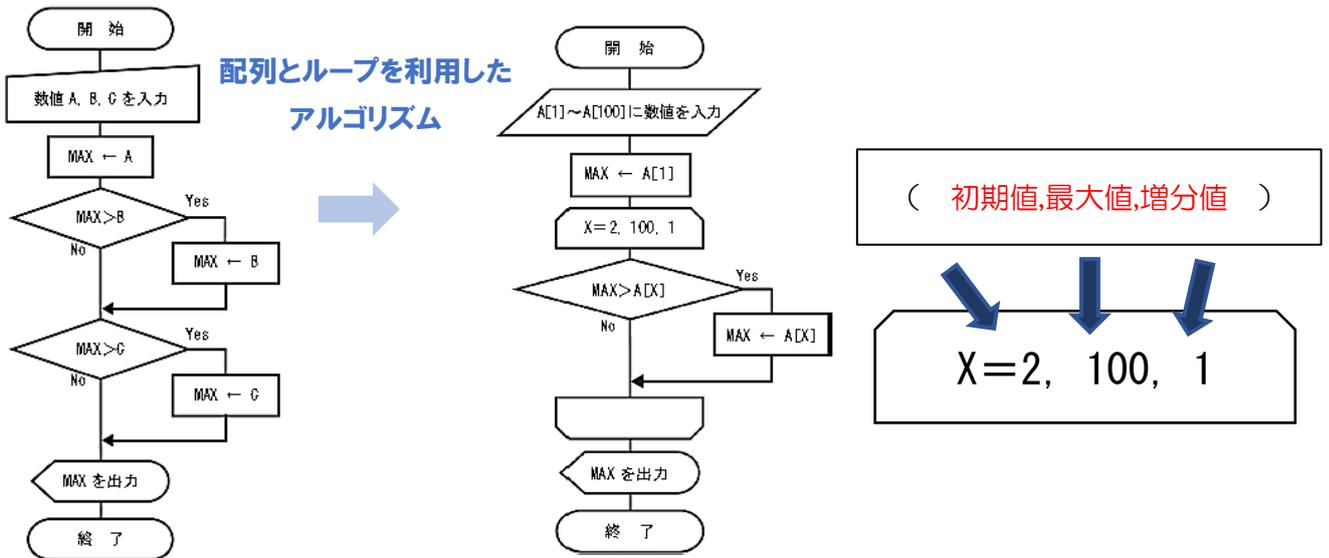
(3) いろいろなアルゴリズム

① 最大値を求める

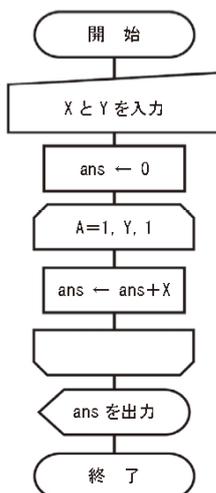
方法1 組み合わせで比較



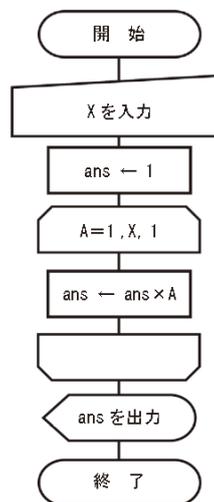
方法2 最大値を更新



② べき乗を求める



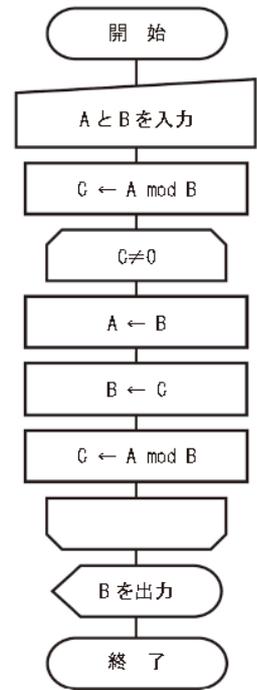
③ 階乗を求める



④最大公約数を求める

⇒ (ユークリッドの互除法)

- ① $1071 \div 1029 = 1 \dots 42$
 - ② $1029 \div 42 = 24 \dots 21$
 - ③ $42 \div 21 = 2 \dots 0$ ←余りが0になったら終了
- ↑
最大公約数となる



問題. 3465 と 7812 の最大公約数を求めましょう。

- $7812 \div 3465 = 2 \dots 882$
- $3465 \div 882 = 3 \dots 819$
- $882 \div 819 = 1 \dots 63$
- $819 \div 63 = 13$ ←割り切れた

答え 13

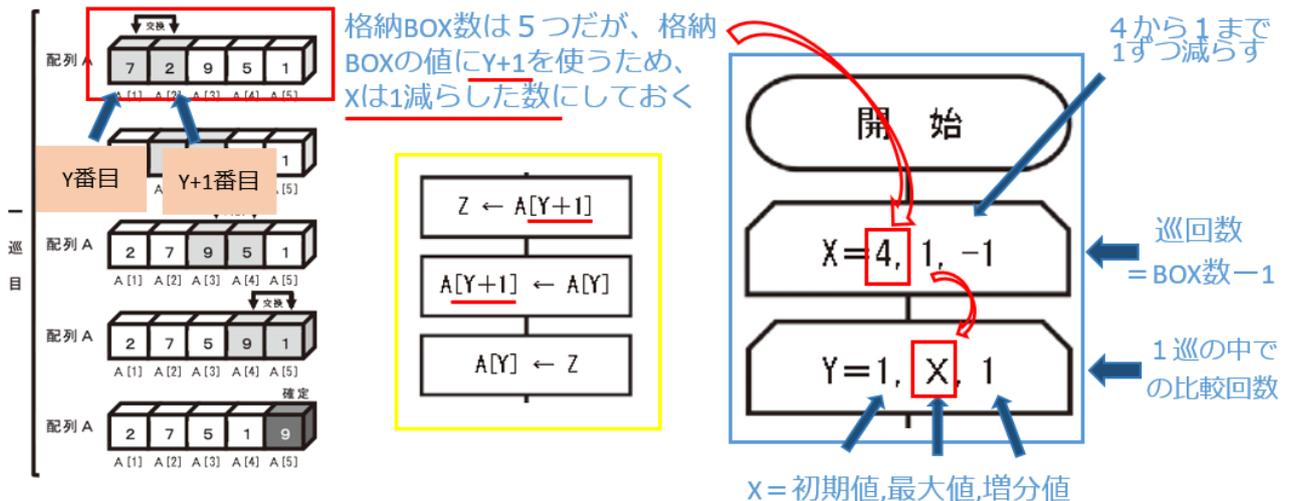
⑤ソート（整列）アルゴリズム

ソート

データを小さい順（昇順）あるいは大きい順（降順）に（ 並べ替える ）作業

バブルソート（基本交換法、隣接交換法）

データを順番に比較し、（ 大小が逆であれば入れ替えを行う ）ソート



問題. P103 図 3-86 の一巡目のデータを流れ図に入れてみよう。