
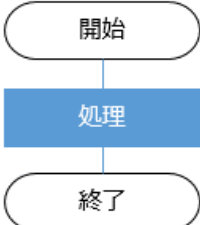


図記号


① ( ) 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 ( ) に配置する必要がある




② ( ) 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、 ( )




③ ( ) 「 ( ) 」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。



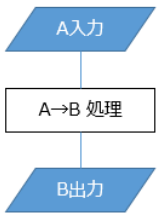
④ ( ) キーボードや ( ) 経由のデータの手動入力




⑥ ( ) ディスプレイへの表示を表す




⑤ ( ) 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”、” ( | ) ”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。



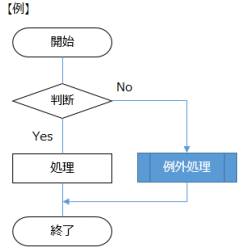
⑦ ( ) 書類の入出力を表す。書類の入力例：レポート、メール、発注書の受領など 出力例：プレゼンテーション、メモ、通信文の作成



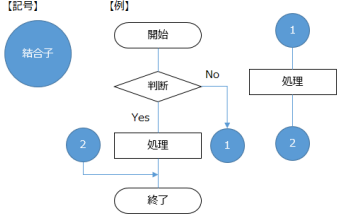
⑧ ( ) 何度も繰り返して行う処理を表す。台形のような六角形の図形で、繰り返し条件と繰り返し終了条件を表す際に配置。ボックスの中に繰り返し名と条件を記入し、間に ( ) を挟む。



⑨ ( ) 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。



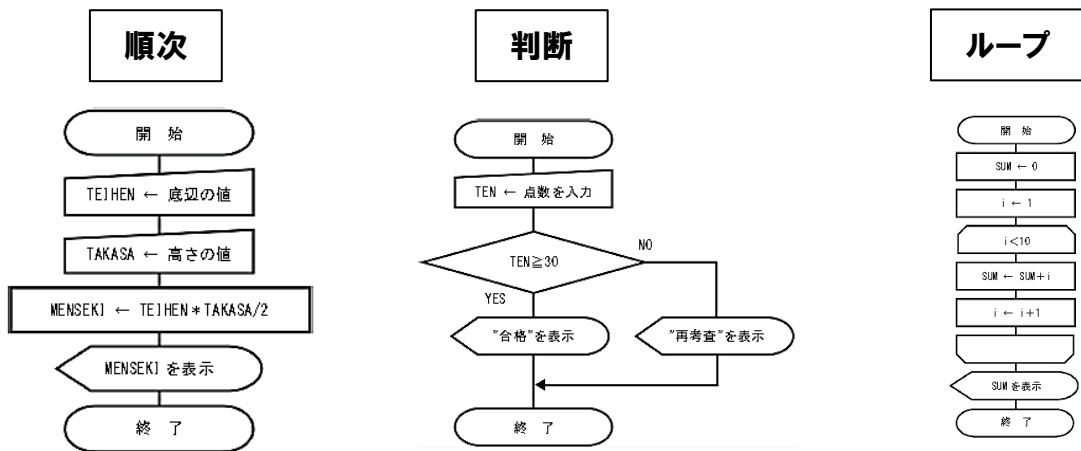
⑩ ( ) フローチャートが長くなったり、複雑になったりした際に同じフローチャート内で ( ) に飛ばす場合に配置する。結合子の中に参照先を記載することが多い。



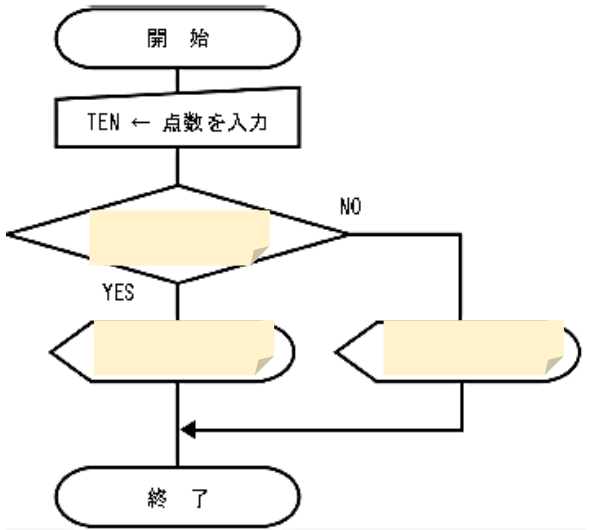
**図記号クイズ!**



(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。

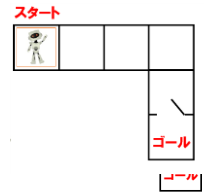
- ① 入口(開始)は1つ
- ② 出口(終了)は1つ
- ③ 線は上から下へ書く
- ④ なるべく判断をいれる
- ⑤ 分岐するときは右に行く

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

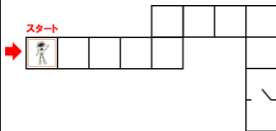
ヒント

- ①
- ②

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  
するまでを流れ図で描きましょう



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。



ルール

使用できる処理は次の4つ。

- ① 前に1マス進む
- ② 右に90° 向く
- ③ 左に90° 向く⇒使用禁止
- ④ ドアを開ける

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。

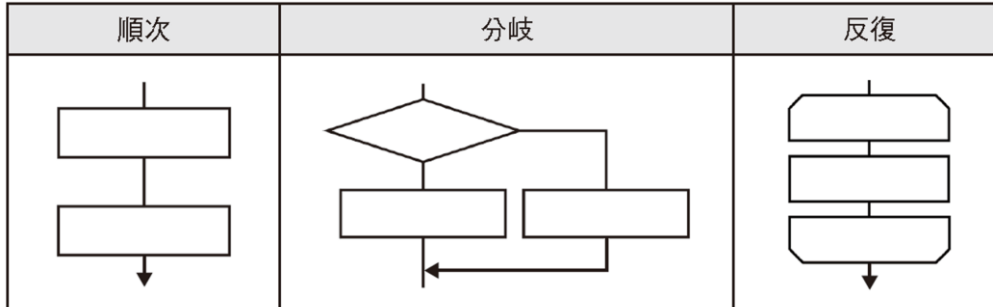
ルール

- 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？

( ) するための ( ) や ( )

(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なさい。

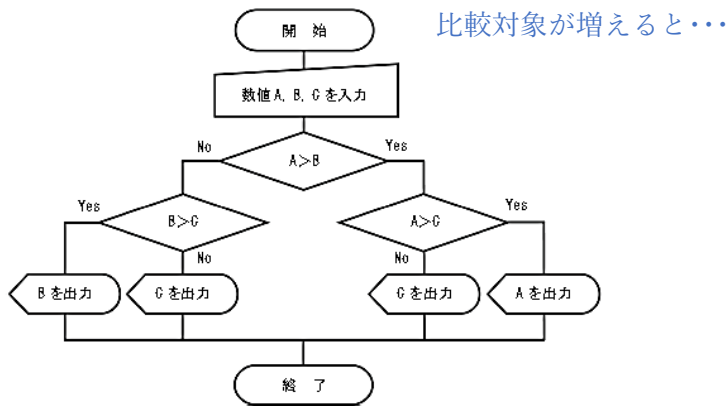
ヒント

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

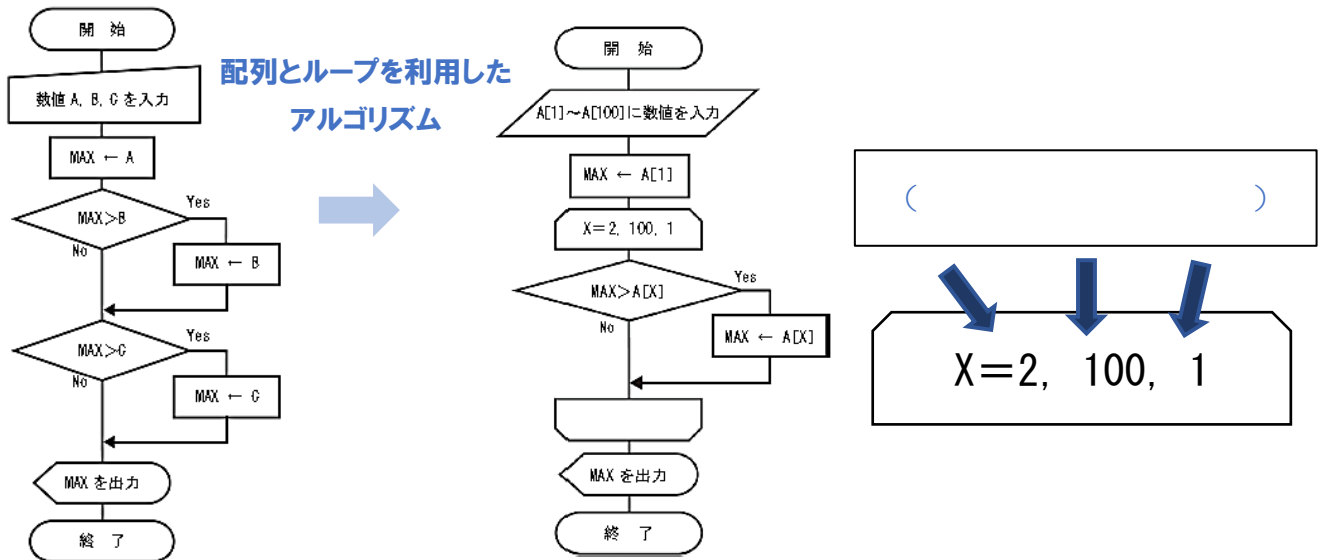
### (3) いろいろなアルゴリズム

#### ① 最大値を求める

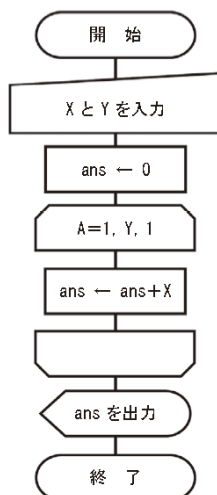
##### 方法1 組み合わせで比較



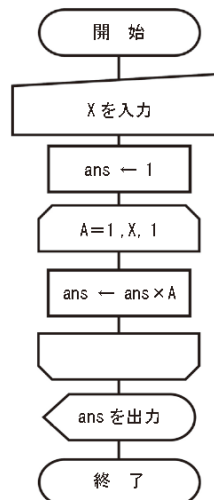
##### 方法2 最大値を更新



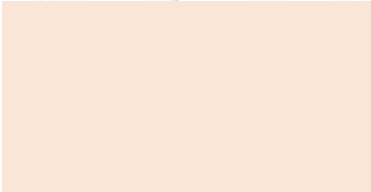

#### ② べき乗を求める

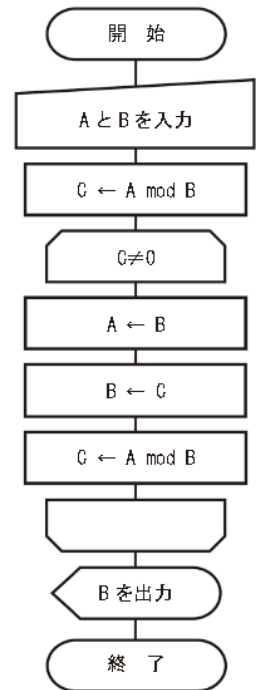


#### ③ 階乗を求める



④最大公約数を求める

- ⇒ ( )
- ①  $1071 \div 1029 = 1 \dots 42$
- ② 
- ③  ←余りが0になったら終了



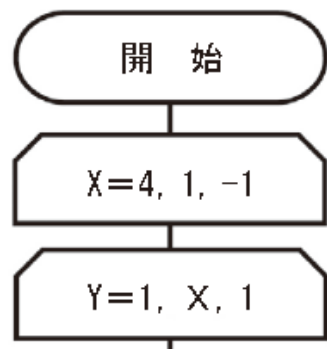
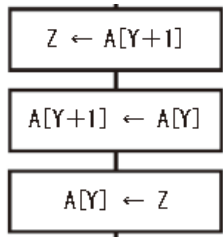
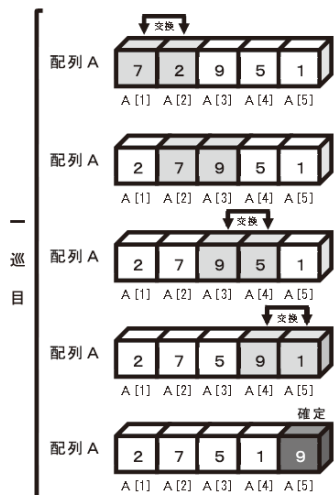
問題. 3465 と 7812 の最大公約数を求めてみましょう。

答え

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（ ）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、（ ）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。