

単元 1

アルゴリズムの基本

変数と演算子

＊ 学習内容

変数、代入、流れ図（フローチャート）、データの交換、算術演算、余り

■ アルゴリズムとは？

私たちは生活の中で、さまざまな手順（アルゴリズム）を利用しています。

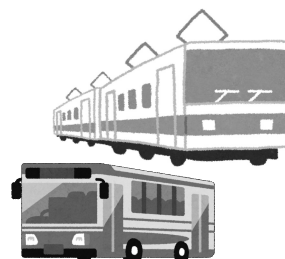
例えば、料理のレシピ、ゲームの攻略法、目的地までの経路案内などは、きちんと筋道すじみちを立てて考えられた手順（アルゴリズム）といえます。



料理のレシピ



ゲームの攻略法



目的地までの経路案内

それでは、日常生活の例として、朝起きてから家を出るまでの手順（アルゴリズム）を考えてみましょう。

<朝起きてから家を出るまでに行うこと>

1. 時計のアラームが鳴ったら、目を覚ます。
2. 時計のアラームを止める。
3. 朝食を食べる。
4. 歯をみがく。
5. 顔を洗う。
6. 着替える。
7. 靴をはく。
8. 家を出る。



この手順（アルゴリズム）にしたがって行動すれば、出掛ける支度しごとくを効率よく進めることができますね。

ところで、先程の手順を次のように入れ替えたら、どうなるでしょうか？

<手順「時計のアラームを止める」を入れ替えた場合>

1. 時計のアラームが鳴ったら、目を覚ます。
2. 朝食を食べる。
3. 歯をみがく。
4. 顔を洗う。
5. 着替える。
6. 靴をはく。
7. **時計のアラームを止める。**
8. 家を出る。



手順「時計のアラームを止める」は、目を覚ました直後に行います。出掛ける直前まで時計のアラームを鳴らし続けると、近所迷惑になるかもしれません。

<手順「朝食を食べる」を入れ替えた場合>

1. 時計のアラームが鳴ったら、目を覚ます。
2. 時計のアラームを止める。
3. 歯をみがく。
4. 顔を洗う。
5. 着替える。
6. 靴をはく。
7. 家を出る。
8. **朝食を食べる。**

マナー違反!?



手順「朝食を食べる」は、歯をみがく直前に行くのが一般的です。歩きながら朝食を食べると、マナー違反と見なされるかもしれません。

このように、手順（アルゴリズム）はその順序がとても重要な意味をもっています。順序を間違えると、期待した結果にならなかつたり効率が悪くなつたりします。

本書では、**データ処理のアルゴリズム**を学習します。

このアルゴリズムには、**データの合計**や**平均**を求めたり、**データの探索**（探す、**サーチ**）や**整列**（並べ替える、**ソート**）を行うものがあります。また、同じ結果になるアルゴリズムが複数用意されている場合があります。

各単元のポイント解説と問題演習から、各アルゴリズムの特長をしっかりと理解するように心掛けましょう。

ポイント 解説①



アルゴリズムの中でデータを一時的に記憶させるには、**変数**と呼ばれる箱を利用します。

また、複数の変数を同時に利用する場合がありますので、各変数にわかりやすい名前を付けてください。

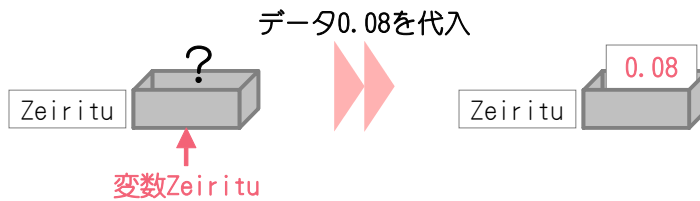
■変数

変数は一つのデータを記憶できます。また、データが記憶された変数に別のデータを**代入**すると、データを書き換えることができます。

【例1】 変数Tankaにデータ500を記憶する場合



【例2】 変数Zeirituにデータ0.08を記憶する場合



【例3】 元のデータ0.08が記憶された変数Zeirituに新しいデータ0.10を記憶する場合



データの種類



- ・整数 : 634、123、0、-999 など
- ・実数 : 0.10、0.08、0.0、-999.9 など
- ・文字列 : "Tokyo"、"東京"、"とうきょう"、"AD2020" など





※文字列データは " 記号（ダブルクォート）で囲んで指定します。

■ 流れ図（フローチャート）

アルゴリズムは、**流れ図**（**フローチャート**）と呼ばれる図式^{ずしき}で表します。

流れ図は、日本工業規格（JIS）^{きかく}で決められた記号^{じごう}を利用して書きます。また、流れ図の記号は、上から下に・左から右に向かって並べます。

<流れ図の記号>

記号	名称	意味
	端子 ^{たんし}	流れ図の開始と終了を示す。 記号の中に「はじめ」や「おわり」を書く。
	準備	データの準備を示す。 変数に初期データを記憶する場合は、右向き矢印で表す。
	処理	データの処理を示す。 変数にデータを記憶する場合は、右向き矢印で表す。
	線	流れ図の流れを示す。 線記号以外の記号をつなぐ。 また、流れの向きを示す場合は、記号に矢先 ^{やさき} をつける。

【例4】 変数Tankaにデータ500を、変数Zeirituにデータ0.08を記憶する場合



【例5】 元のデータ0.08が記憶された変数Zeirituに新しいデータ0.10を記憶する場合



アルゴリズム

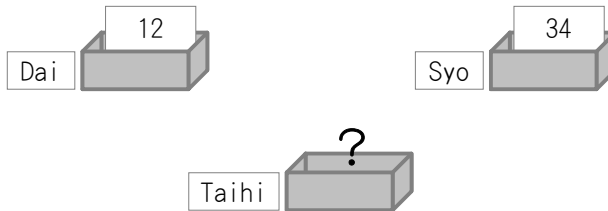
データの交換

データを入れ替えることを**データの交換**といいます。

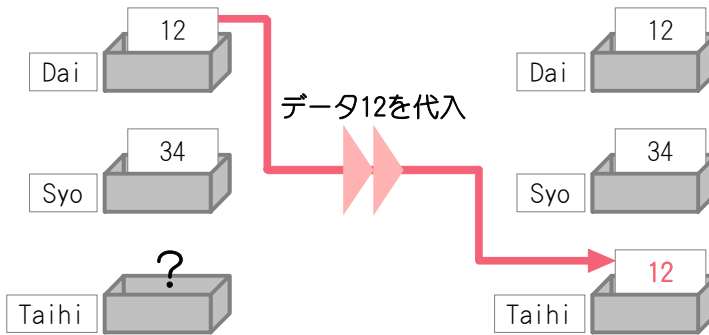


ここでは、変数Daiのデータと変数Syoのデータを、別の変数Taihiを利用して入れ替える手順を説明します。

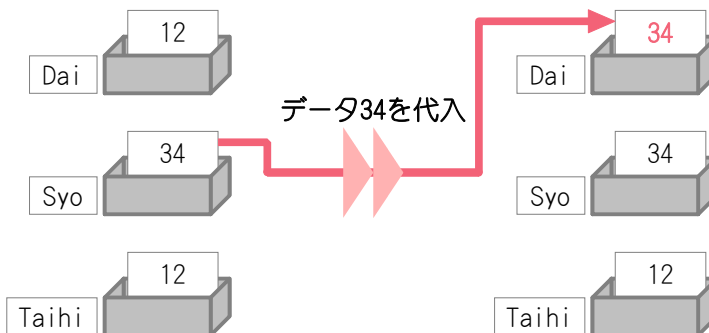
(実行前)



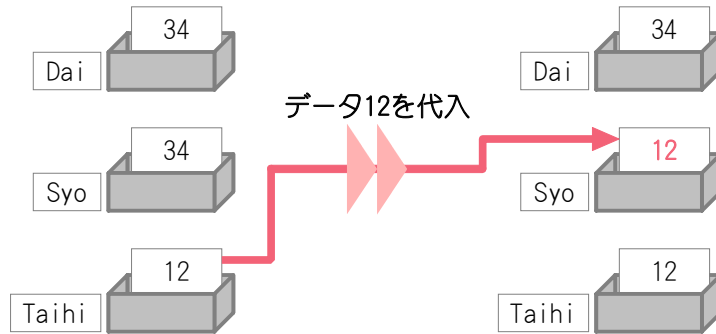
【手順①】 変数Daiのデータ12を変数Taihiに代入します。



【手順②】 変数Syoのデータ34を変数Daiに代入します。



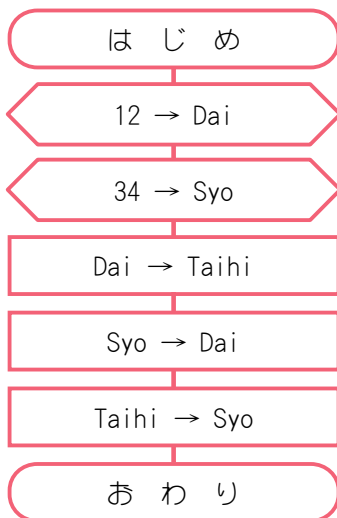
【手順③】 変数Taihiのデータ12を変数Syoに代入します。



流れ図（フローチャート）をたどって、どのような手順で処理が行われているか確認することを**トレース**といいます。

データを交換する手順を表した流れ図（フローチャート）中の各変数がどのように更新されていくかを、次の**トレース表**を活用して確認しましょう。

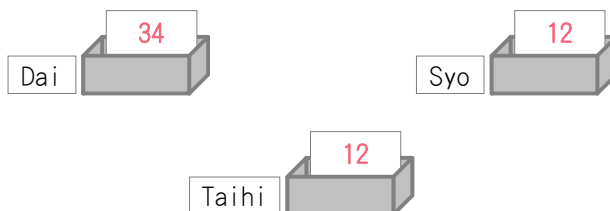
<流れ図>



<トレース表>

	変数Dai	変数Syo	変数Taihi
...	12	?	?
...	12	34	?
...	12	34	12
...	34	34	12
...	34	12	12

(実行後)

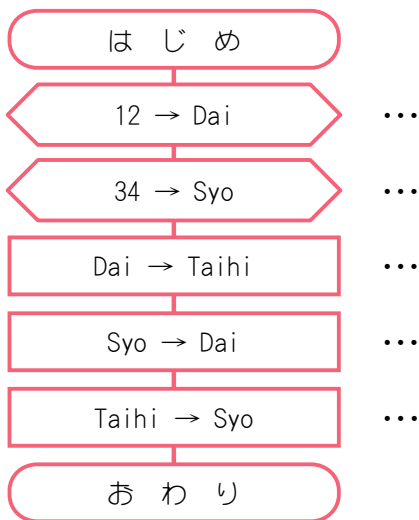




問題 1

次の流れ図を実行した場合のトレース表を完成しなさい。なお、変数のデータ不定は「？」で表します。

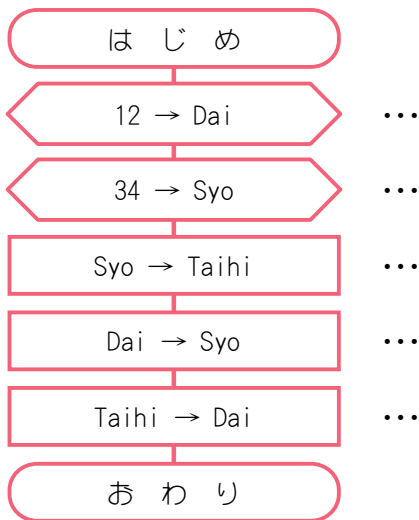
<流れ図①>



<トレース表①>

変数Dai	変数Syo	変数Taihi

<流れ図②>



<トレース表②>

変数Dai	変数Syo	変数Taihi

問題 2

次の手順①～④を読んで、流れ図を完成しなさい。

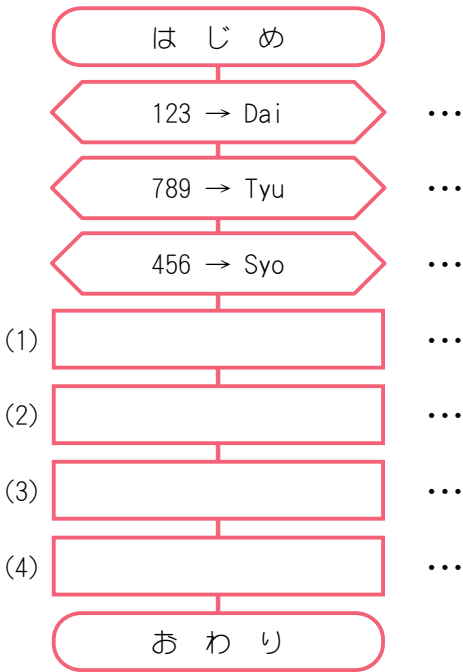
【手順①】 変数Daiのデータ123を変数Taihiに代入します。

【手順②】 変数Tyuのデータ789を変数Daiに代入します。

【手順③】 変数Syoのデータ456を変数Tyuに代入します。

【手順④】 変数Taihiのデータ123を変数Syoに代入します。

<流れ図>



<トレース表>

	変数Dai	変数Tyu	変数Syo	変数Taihi
...	123	?	?	?
...	123	789	?	?
...	123	789	456	?
(1) ...	123	789	456	123
(2) ...	789	789	456	123
(3) ...	789	456	456	123
(4) ...	789	456	123	123