情報の授業

アルゴリズムとプログラム Ver. 3.0

情報の授業の中では、やや難しいアルゴリズムとプログラムの学習です。
動画、スライド、実習の組み合わせですが、自分のペースで進めてください。
わからない時は、まわりの友達に相談するか、先生に聞いてください。

Go.Ota, Aug.2020

学習の進め方(1)

理解

- 今見ている、この資料に学習内容の説明・課題などあります。個人のペースで学習を進めてください。各学習が終ったら、次のスライドを見てチェックシートにチェックを入れてください。
- 課題の中には簡単なものもあり、すでにわかっている人は飛ばしてやってもいいです。
- 授業中は必要だったら歩き回って、友達に相談したり、友達に教えてあげていいです。
- 分からないことがあったら、先生に質問して聞いてもいいです。

学習の進め方(2)

理解

	No	内容	スライドNo	チェック			
				[動画]	[理解]	[打ち込み]	[開発]
ここまでは、やってみよう。	1	学習の進め方	2		□		
	2	準備運動 1: Scratch の四則演算	4		□	□	
	3	準備運動 2: ネコに自分の名前を言わせる	5		□	□	
	4	プログラムと変数	6	□			
	5	課題 1: 入力した 2 個の数で四則演算	7				□
	6	打ち込み 1: 合格判断	8		□	□	
	7	課題 2: 合格不合格判断	9				□
	8	課題 3: 3 つの数の合計	10				□
	9	課題 4: 正三角形の判断	11				□

・スライドNo. このスライド内の番号
・動画 :指定された動画をみたらチェック
・理解 :スライドの内容を見て自分なりにわかったらチェック
　　スタジオ内のプログラムの意味がわかったらチェック
・打ち込み：スライドのプログラムを入力して動作確認
　(簡単なプログラムで打ち込む必要なしと判断した場合は打込まなくていいです)
・開発：スライドの課題のプログラムを作って、ちゃんと動作したらチェック。

準備運動1: Scratchの四則演算

理解・打込み

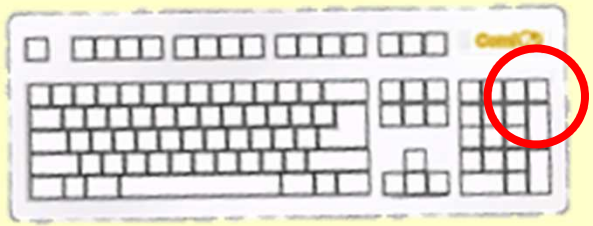
こんな簡単なプログラムは打ち込むのは面倒という人は飛ばして進んでいいです。

まず、準備運動で、四則演算(足し算、引き算、掛け算、割り算)のプログラムを作って動かしてみます。

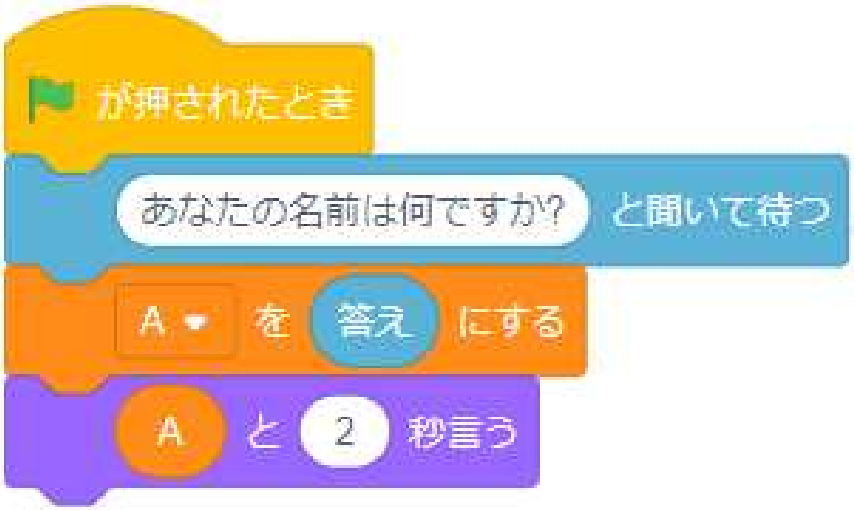
・数値は好きな値をいれてください。
・普通の数学の記号が、プログラムでは違うものがあります。

掛け算　＊
割り算　/

(キーボードの右側にテンキーがついている時はこれらの記号があります)



準備運動2: ネコに自分の名前を言わせる 理解・打込み



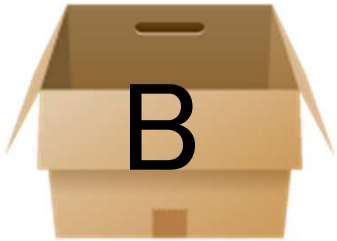
自分の名前をキーボードから入力して、ネコに言わせてみます。

ヒント: Aという名前の変数を作って使います。

こんな簡単なプログラムは打ち込むのは面倒という人は飛ばして進んでいいです。

プログラムと変数

動画

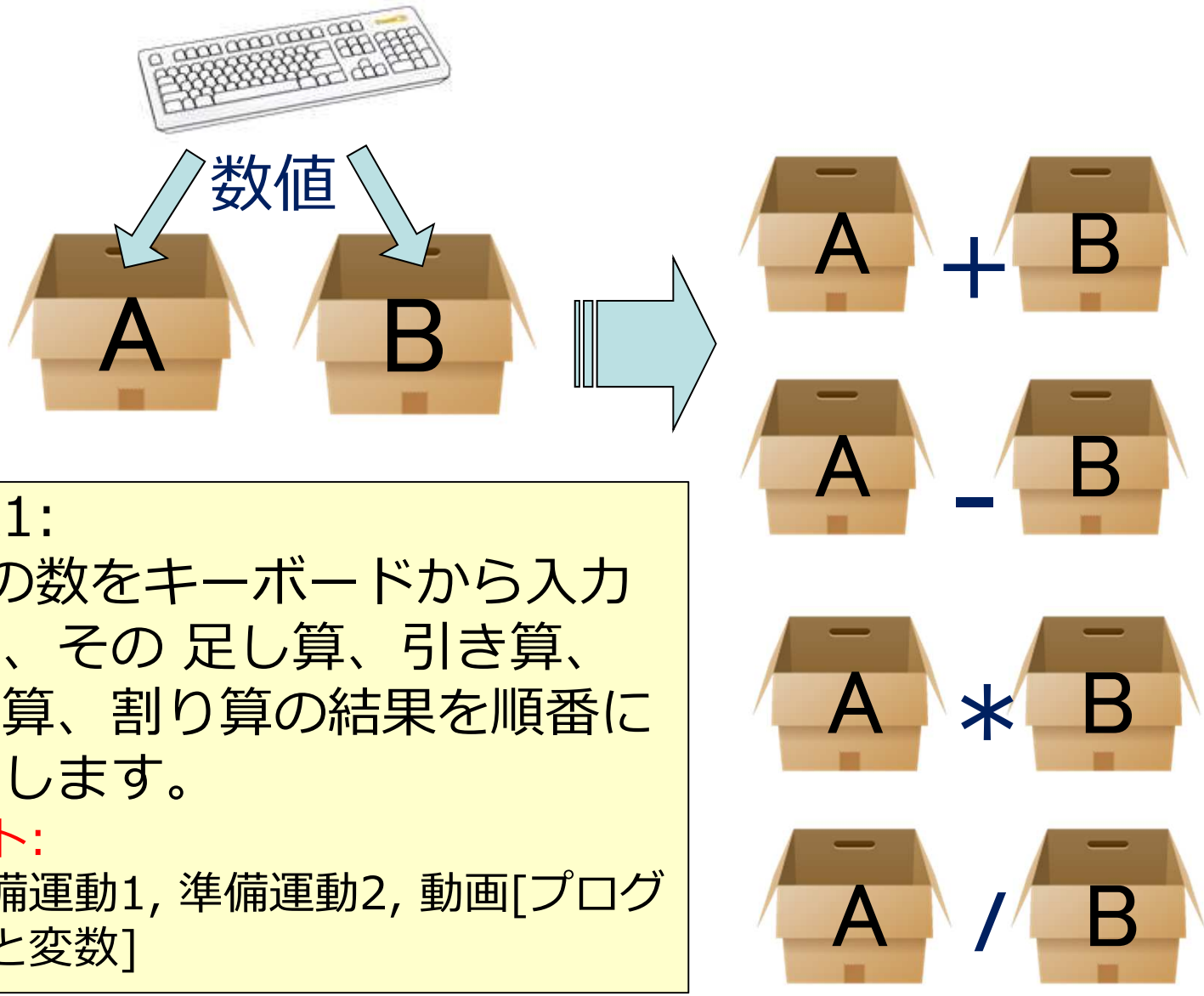


プログラムの中で役に立つ変数についての動画です。次の課題のヒントもありますから、しっかり見てください。

https://youtu.be/-WwvWPreFeo

課題1: 入力した2個の数で四則演算

開発



課題1:
2個の数をキーボードから入力して、その 足し算、引き算、掛け算、割り算の結果を順番に表示します。

ヒント:
・準備運動1, 準備運動2, 動画[プログラムと変数]

打ち込み1: 合格判断

理解・打込み

こんな簡単なプログラムは打ち込むのは面倒という人は飛ばして進んでいいです。

変数Aに数を入力して、70より大きければ(70は入らない)、ネコに合格と言わせてみます。

課題2: 合格不合格判断

開発

変数Aに数を入力して、
もし、70より大きい**なら**(70は入らない)、ネコに合格と言わせます。
そう**でなければ**、ネコに不合格と言わせます。

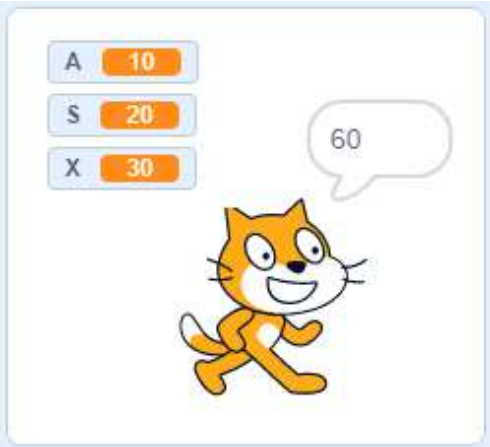
ヒント: プログラムを動かしたあと、次のような数を入れて正しく動作するか確認してみよう。

数の種類	例	ネコ
70よりかなり大きい	90	合格
70より少し大きい	71	合格
70	70	不合格
70より少し小さい	69	不合格
70よりかなり小さい	30	不合格

課題3: 3個の数の合計

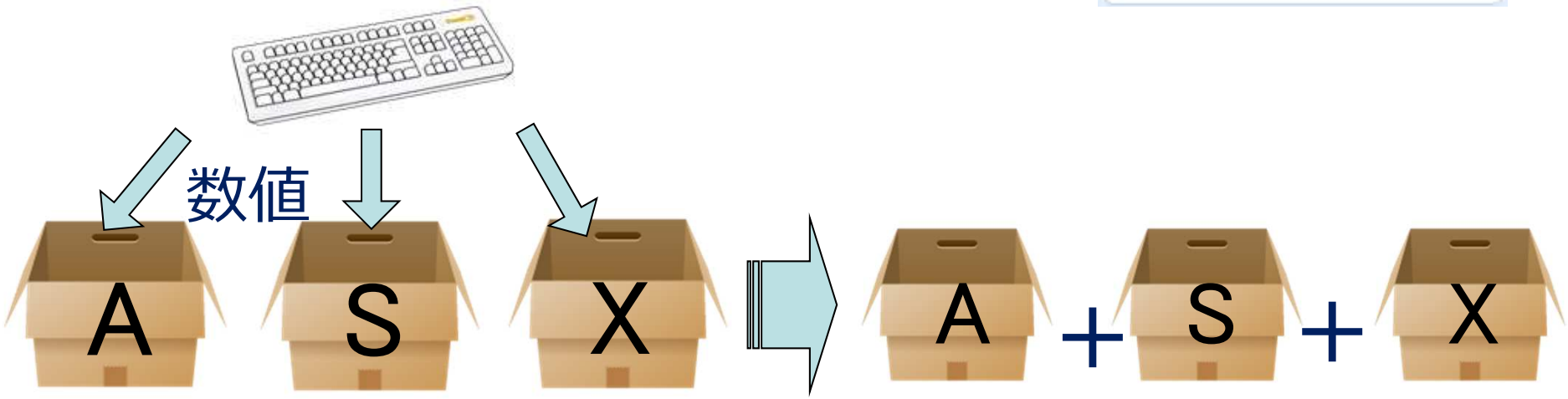
開発

3個の数をA, S, Xの3つの変数に入力して合計をネコに言わせるプログラムを作成してください。



ヒント:

課題1の復習みたいなプログラムです。



課題4: 正三角形の判断

開発

A, S, Xの3個の三辺の値を入力して、すべての値が同じ場合に、「正三角形」、そうでない場合は「正三角形じゃない」と表示するプログラムを作ってみよう。

ヒント: プログラムを動かしたあと、次のような数を入れて正しく動作するか確認してみよう。

数の組み合わせ	A	S	X	ネコ
すべてが同じ	10	10	10	正三角形
全部違う	10	20	30	正三角形じゃない
AとSだけ同じ	10	10	20	正三角形じゃない
SとXだけ同じ	20	10	10	正三角形じゃない

打ち込み2: 1から10までの数を言う

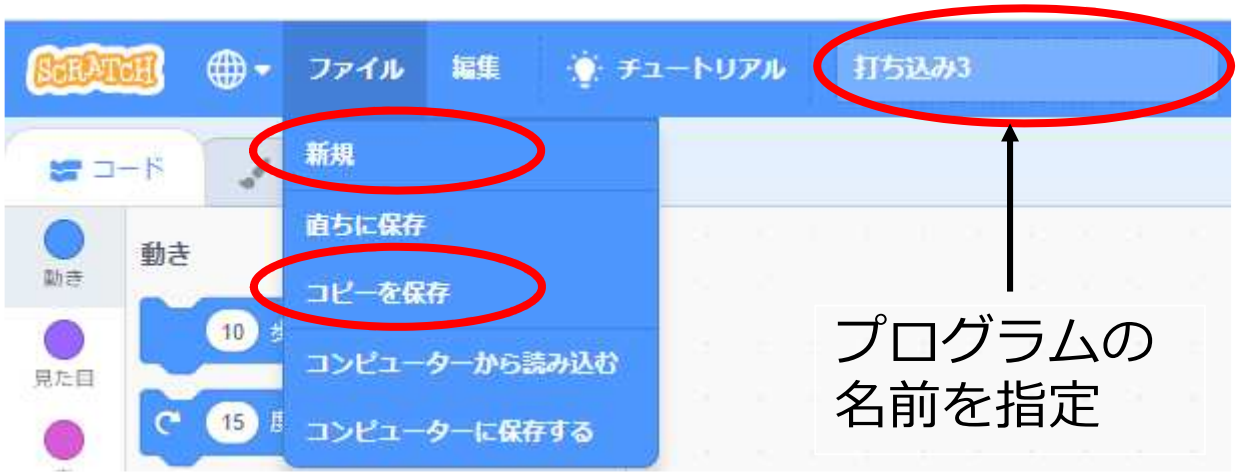
理解・打込み

こんな簡単なプログラムは
打ち込むのは面倒という人
は飛ばして進んでいいです。

1,2,3,・・・,10までの数を
ネコと順番にと言わせてみます。

プログラムに名前をつけて保存

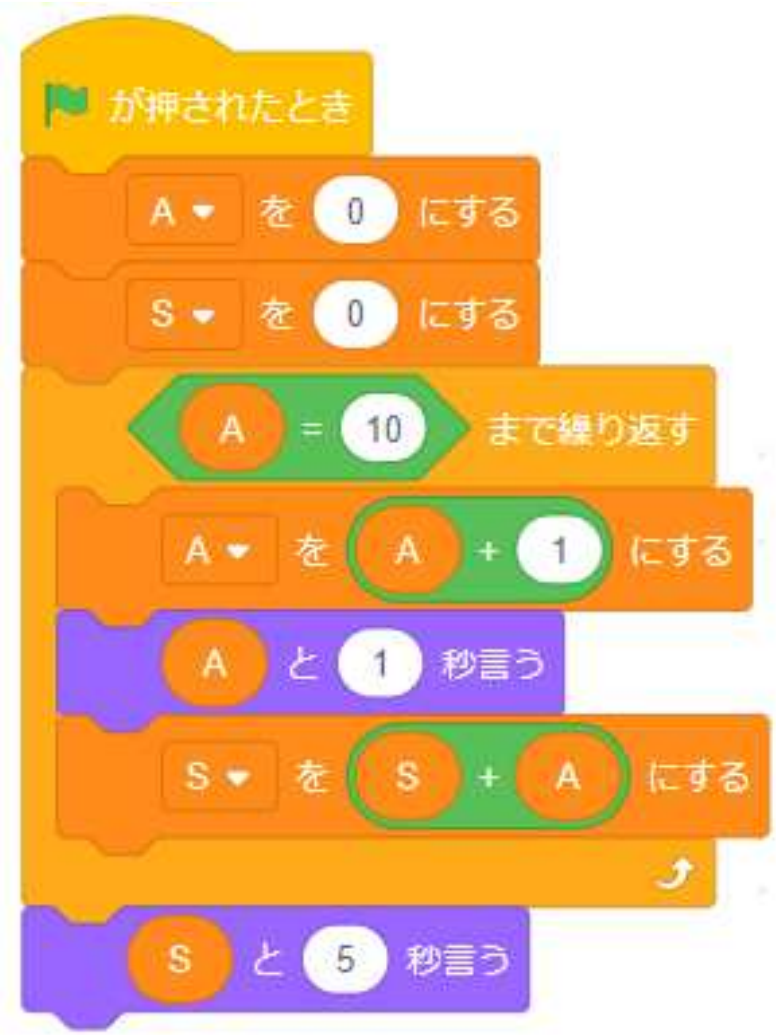
理解

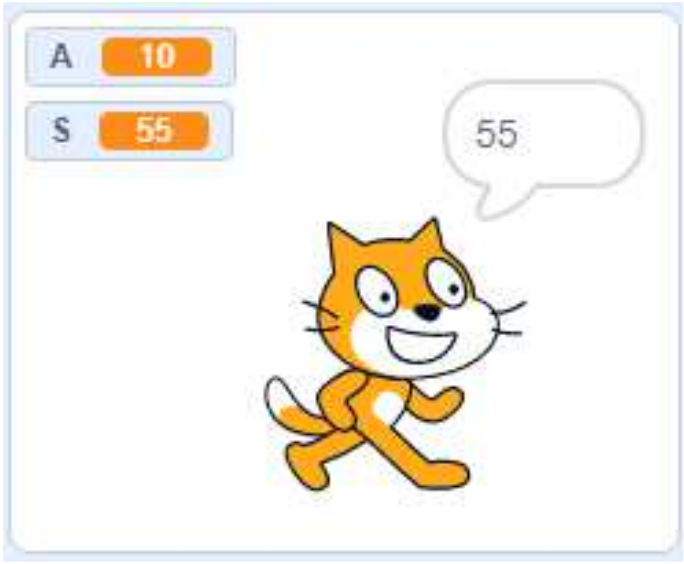


これからの課題や打ち込むプログラムは前に作ったものを改造すると楽なものが多くなります。
個々のプログラムにつけて保存したり、コピーを保存で流用したりしましょう。

打ち込み3: 1から10までの合計

理解・打込み

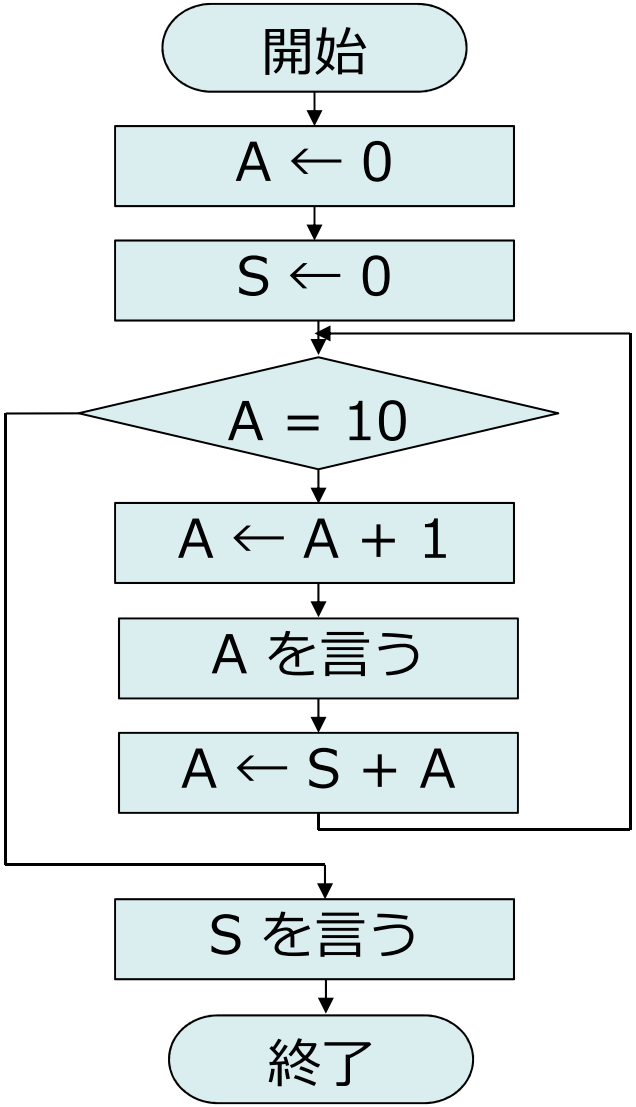




1,2,3,・・・,10までの数をネコが順番に言いながら、その数を足して、最後にその合計を言います。

プログラムとフローチャート

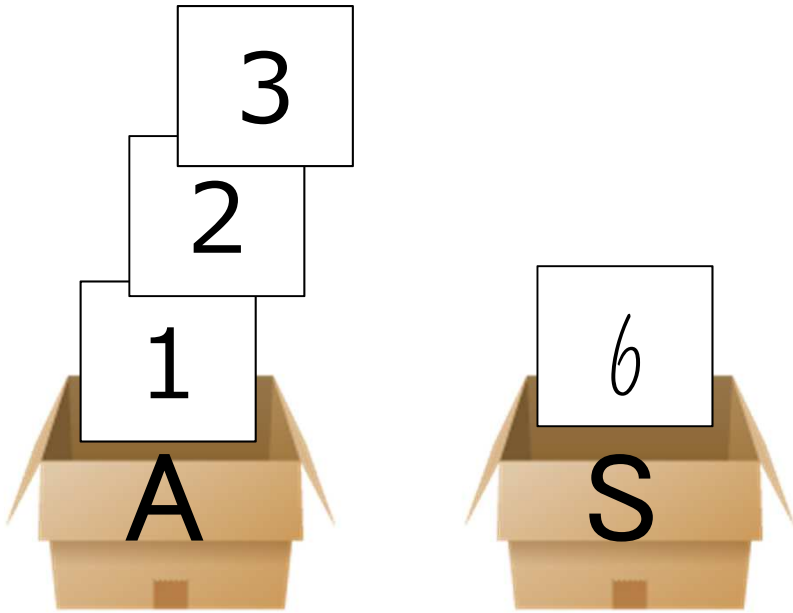
動画



プログラムもだんだん難しくなっていきます。そのため、楽に問題を考えられるようにフローチャートを学習していきます。ちなみに、前に「 1から10までの合計」は左のようになります。
(開発中)

変数/リストシートを使って考えよう

理解



動画で説明したように、段ボールの箱を変数として見立て、変数/リストシートが用意されています。実際のプログラムの動きに合わせて、変数の中がどう変わるか、紙に書いて入れてみましょう。

課題5: 2からXまでの偶数の合計

開発

Xの値を入力して、2からX以下の偶数の合計を求めるプログラムを作成してください。
Xには偶数だけ入力します。

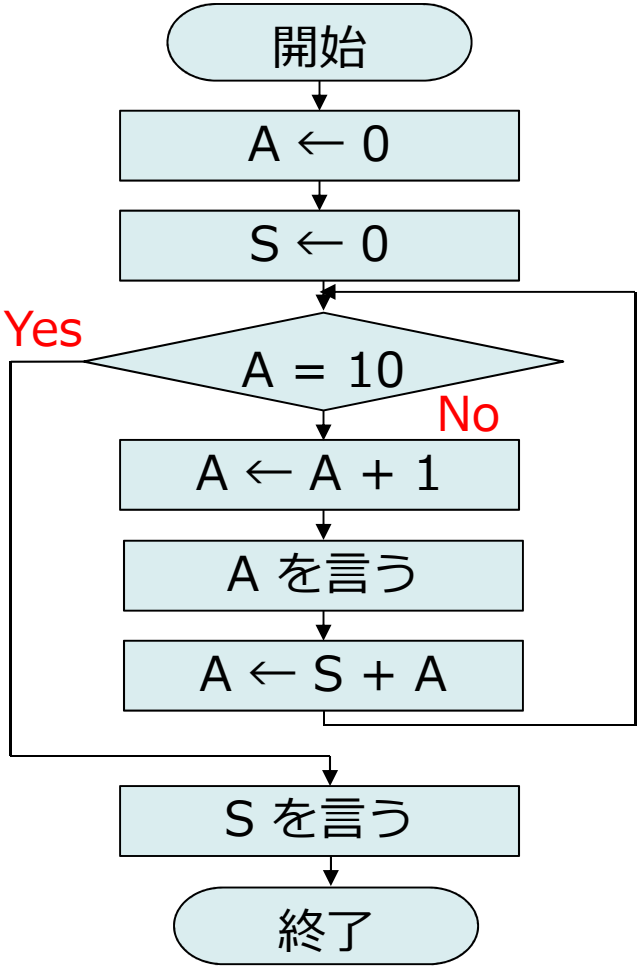
例: X = 6の場合、 2+4+6
　　X = 10の場合、 2+4+6+8+10

ヒント:

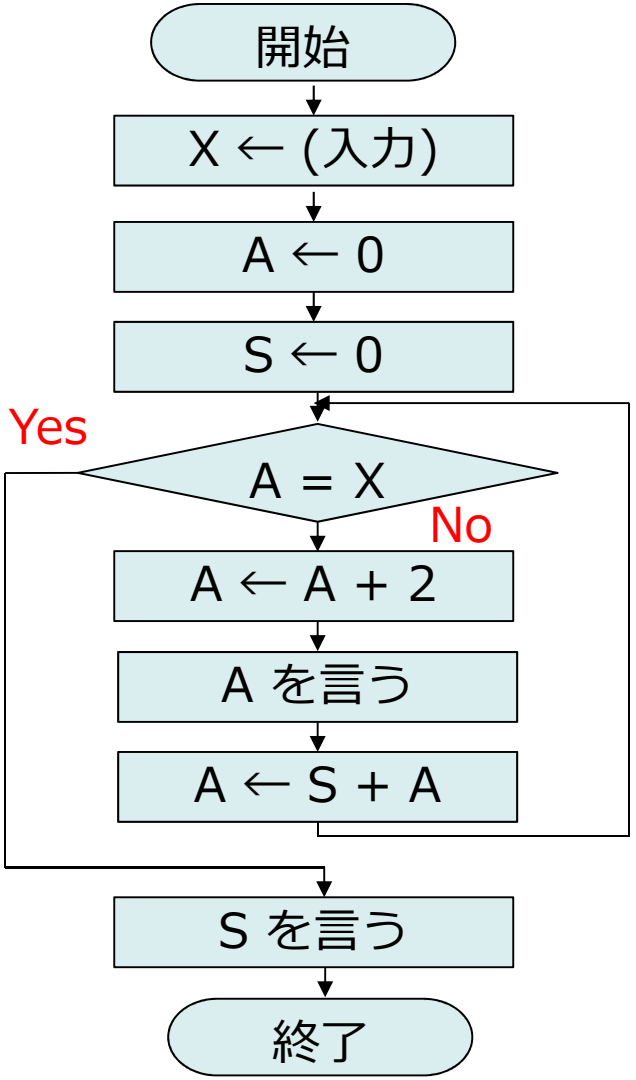
打ち込み3:「1から10までの合計」のプログラムを改良して作ります。まず、2から10までの偶数の合計を求めるプログラムにしてから、Xを入力して、X以下の合計を求めるプログラムを作ると楽かもしれません。

次のスライドのフローチャートも参考にしてください。

課題3補足: 2からX以下の偶数の合計



1から10の合計



2からXまでの偶数の合計

たくさんの数の合計の説明

次からたくさんの数を入力して合計を
求めるプログラムをつくっていきます。

今まで、数を入力するのに変数をつくっていきました。課題3では、A,S,Xの3つの変数を使って合計を計算しました。
ただし、が、たくさんの数を入力するには、A,B,C,D,E・・・・と多くの変数を使うのは大変です。

こんな時にリスト(配列)使うと便利です。
次からリスト(配列)を学習していきます。

打ち込み4:リストを使った3つの数の合計

理解・打込み

ヒント:

リストDの作り方は次のスライドを参考にしてください。

打ち込み4補足: 配列(リスト)を作る

理解・打ち込み

変数の中で[リストを作る]を指定。

リスト名を指定して作成

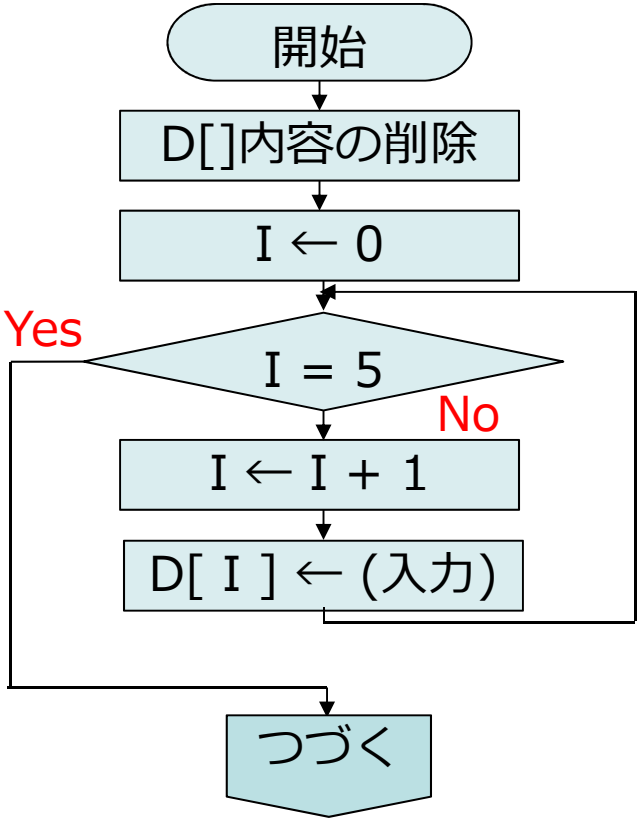
リストはDに番号がついている、さくたんの箱のイメージです。

打ち込み5:リストを使った5つの数の合計

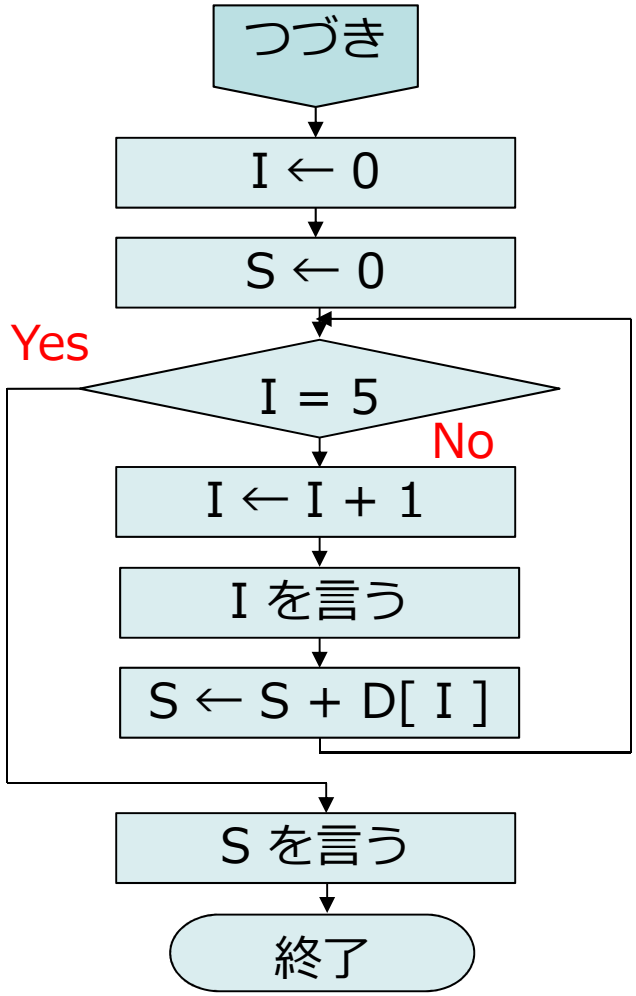
理解・打込み

いきなりプログラムが難しくなりました。
前の打ち込み4のプログラムでも、たくさんの数を入力するのは大変なので、ループを使ってプログラムを効率的に作ったものです。
次のスライドにフローチャートを示します。

打ち込み5補足:リストを使った5つの数の合計



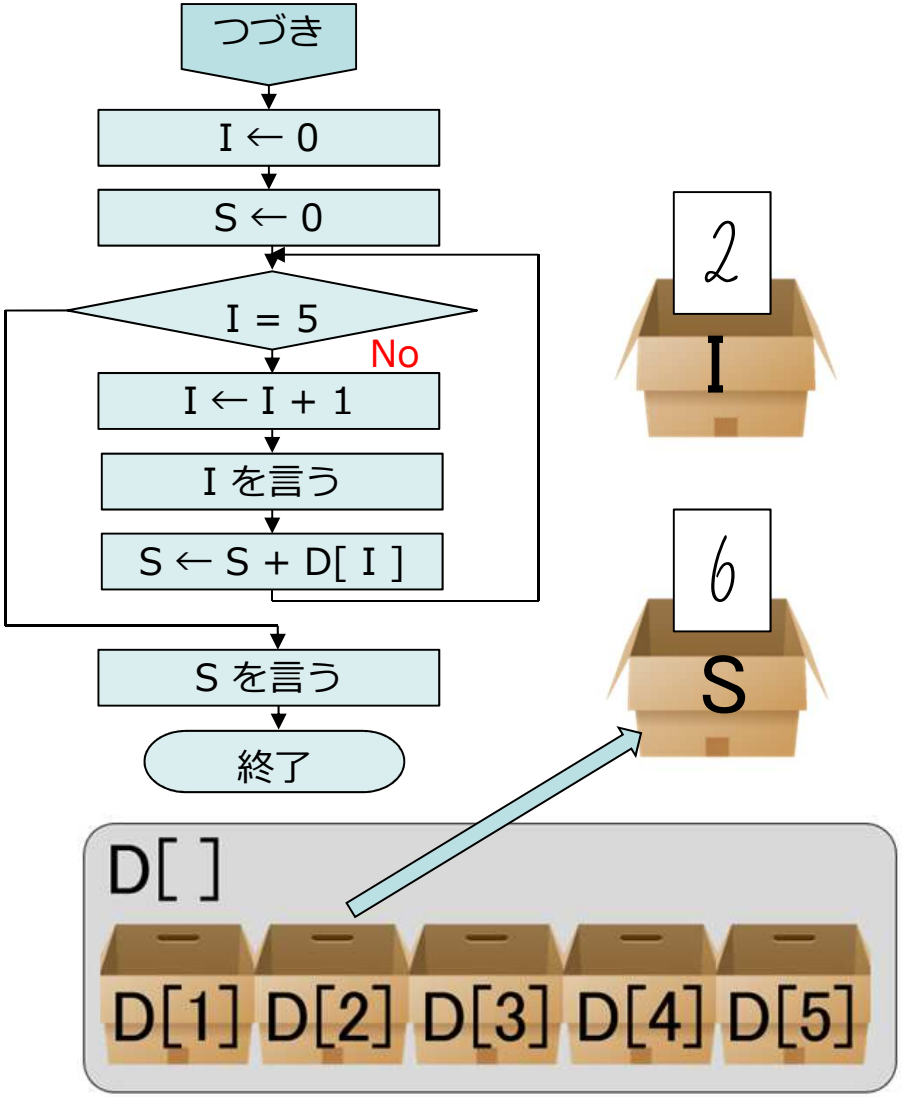
リストDに5個の数を
入力する部分



リストDに入っている5個の
数を合計する部分

リストと繰り返し

動画



リストや繰り返しを使えとプログラムの動きも分かりにくくなってきます。ここでは動画でその動きを説明しています。
(開発中)

最終目標の課題の説明:数の並び替え

予めリストD[1]からD[5]まで数を入れておきます。

この中の数を小さい順番に並び替えてリストD[1]からD[5]に入れなおしてください。

理解

これが最後の目標の課題です。できると思う人は、いきなり開発してもいいです。
少し難しいと思う人は、次の課題を順番にやっていきましょう。

・(休憩): おみくじ
・配列から数を見つける
・配列の中の一番小さい数を見つける

(ここまではがんばる)

・配列の中の一番小さい数を、配列の先頭に入れる。
・二重のループを使う
・並び替えの開発

休憩: おみくじをつくる

理解・打ち込み

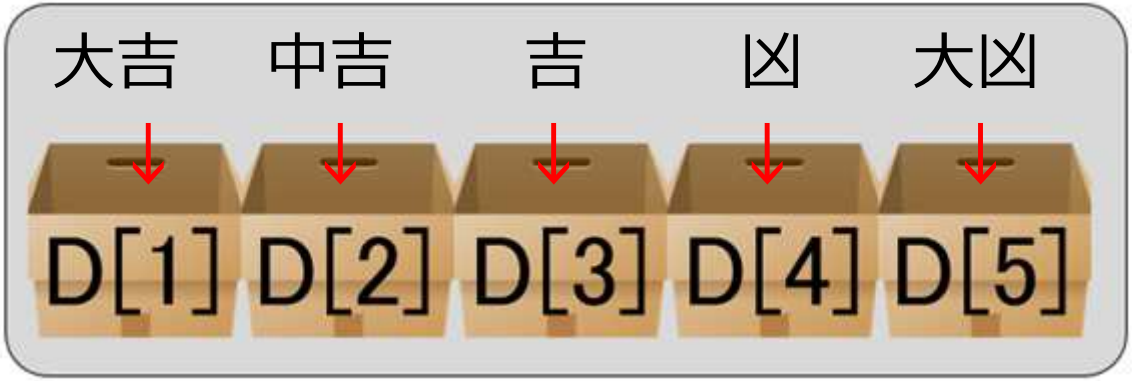
リストを使っておみくじを作ってみます。

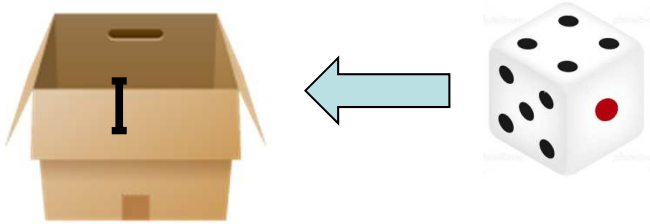
動作の説明は次のスライドにあります。

リストが理解できて、休憩せずにガンガンやりたい人は飛ばして進んでいいです。

「おみくじを作る」の説明

理解・打ち込み





Iに1～5のどれかの数が入る。

D[I] の内容を言う
(Iに入っている数に対応したD[]の内容)

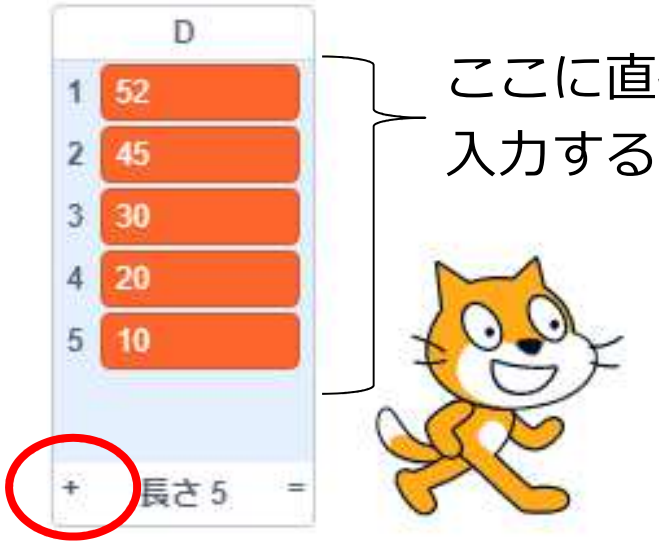
リストに数をセットする方法(1)(2)

理解

リストに数をセットする方法にはいくつかあります。ここでは、
3つの方法を理解しましょう。

(1) 入力する。
スライド29で説明済み

(2) 配列(リスト)に直接、
数値を入力する。



次のスライドで(3)を説明しています。

配列に数をセットする方法(3)

理解・打ち込み

(3) ランダムな数を設定する。

1から100の目の出るサイコロを振って、順番に数値を入れているイメージです。

課題6:リストの中から数を探す(検索)

開発

予めリストD[1]からD[5]まで数を入れておきます。

つぎに変数のAに数を入力して、

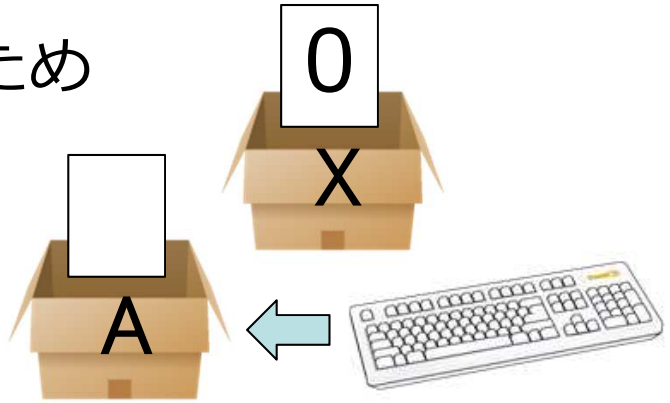
その数がD[1]からD[5]にあれば、何番目に入っていたか変数Xにその番号を入れます。

入っていなければ変数Xに0を入れます。

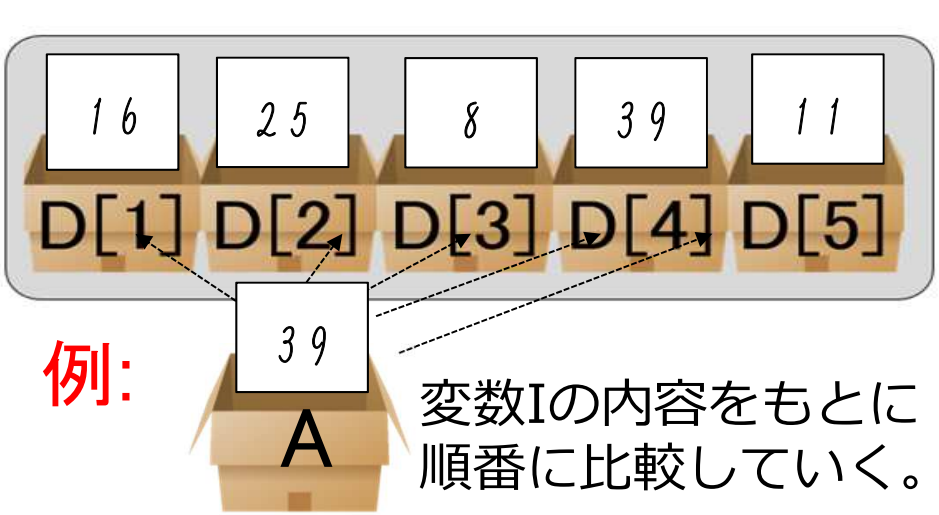
次と次のスライドにヒントがあります。
基本的に、打ち込み5の「リストを使った5つの数の合計」をちょっとだけ改良して作ります。

課題6補足:配列の中から数を探す(検索)の考え方

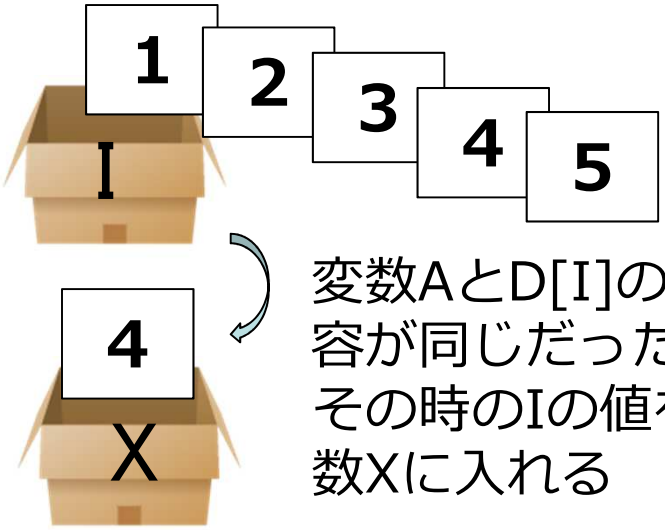
1. あらかじめ配列(リスト)Dに5つの数を入れておきます。
2. 変数Xに見つからなかった時のために0を初めに入れときます。
3. 変数Aに探す数を入力します。



4. 変数Iを1～5まで変えていき、変えるときのD[I]とAの内容を比較します。もし等しければ、I(何番目かの番号)の値をXに入れます。等しいものがなければXは0のままです。

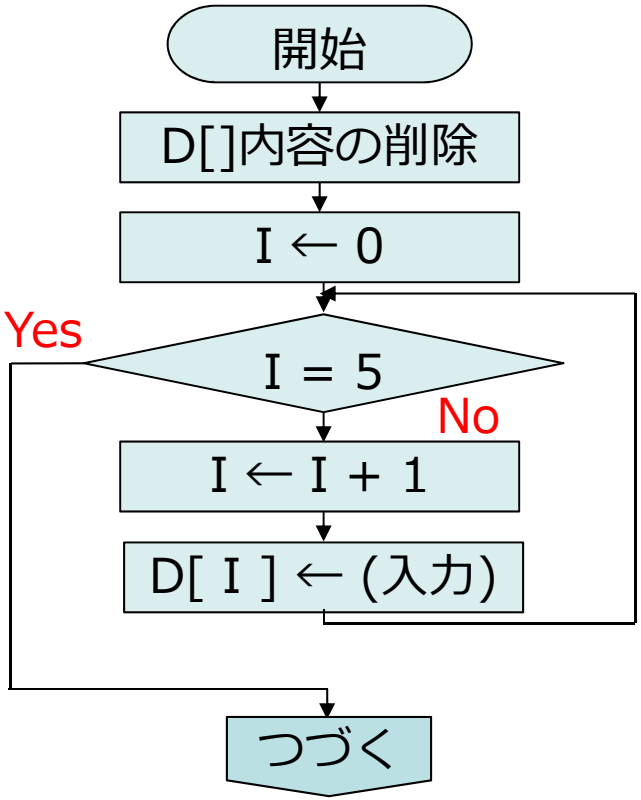


変数Iの内容をもとに順番に比較していく。

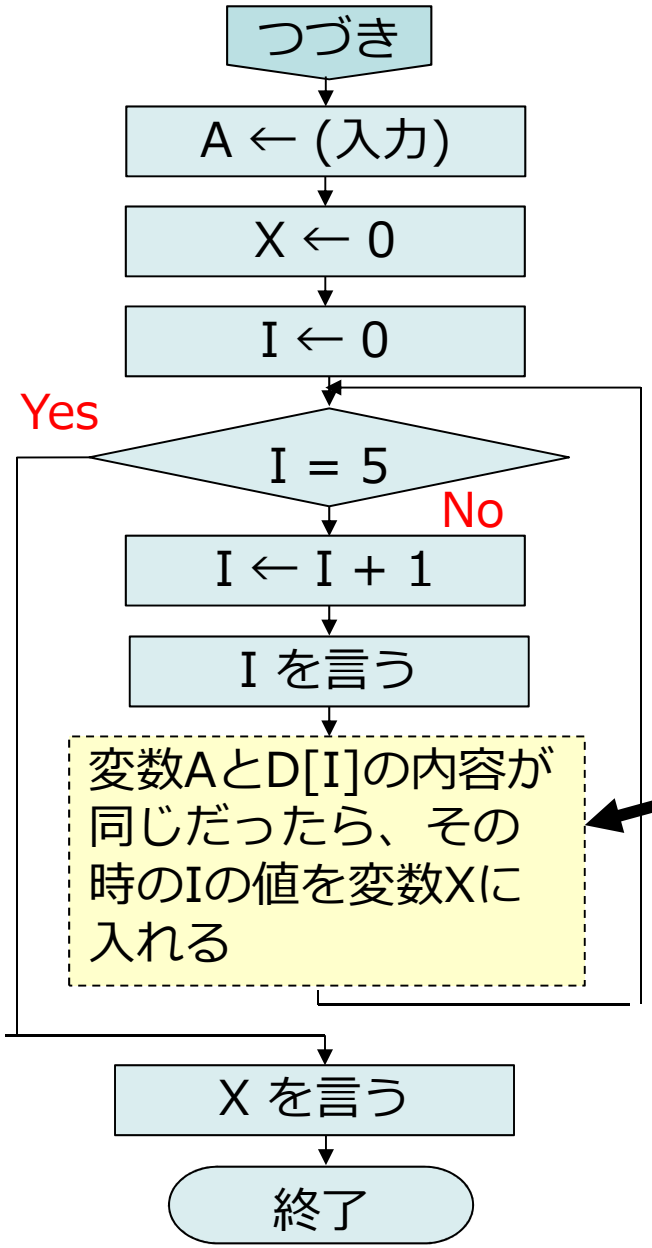


変数AとD[I]の内容が同じだったら、その時のIの値を変数Xに入れる

課題6補足:リストの中から数を探す(検索)のヒント



リストDに5個の数を
入力する部分
(乱数を使って入れても
いいです。)



この部分を考えて、プログラムを完成されてください。

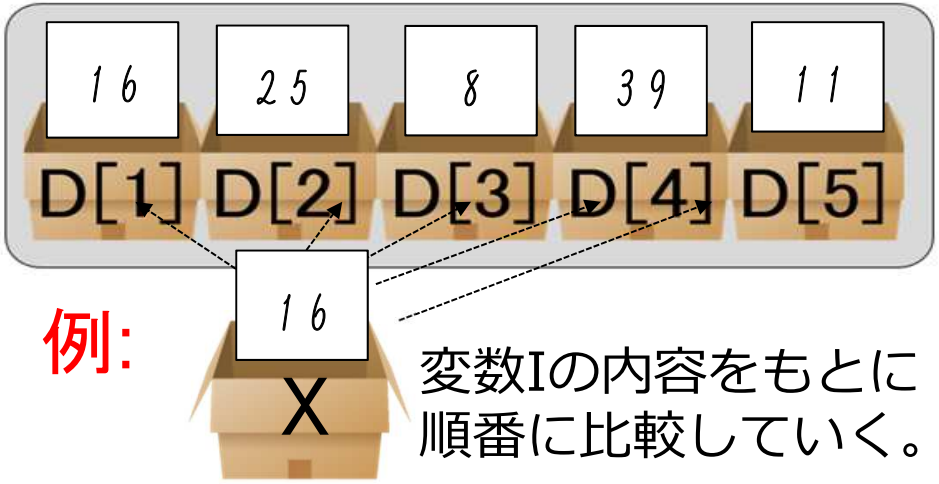
課題7:リストの中の一番小さい数を見つける

予めリストD[1]からD[5]まで数を入れておきます。

その中で一番小さな数をXに入れます。

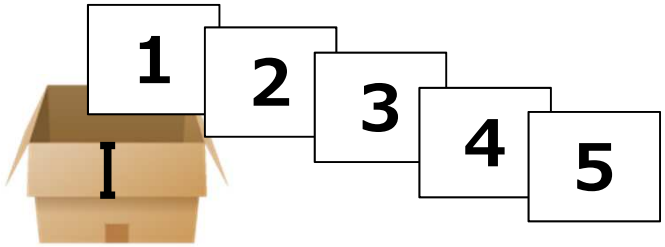
開発

変数Iを1～5まで変えていき、変えるときのD[I]とXの内容を比較します。D[I]がXと小さければ、D[I]の値をXに入れます。



例:

変数Iの内容をもとに順番に比較していく。

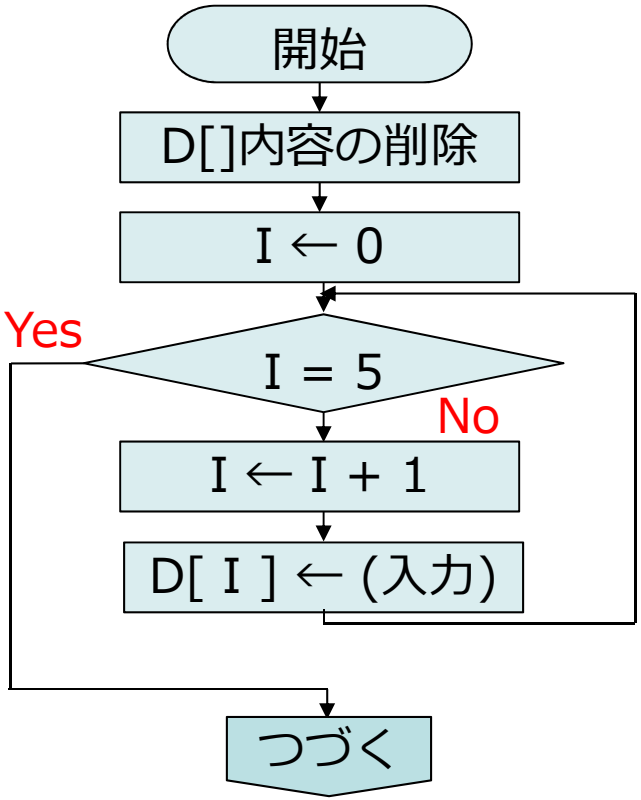


変数XとD[I]の内容を比較してD[I]が小さければ、その値を変数Xに入れる

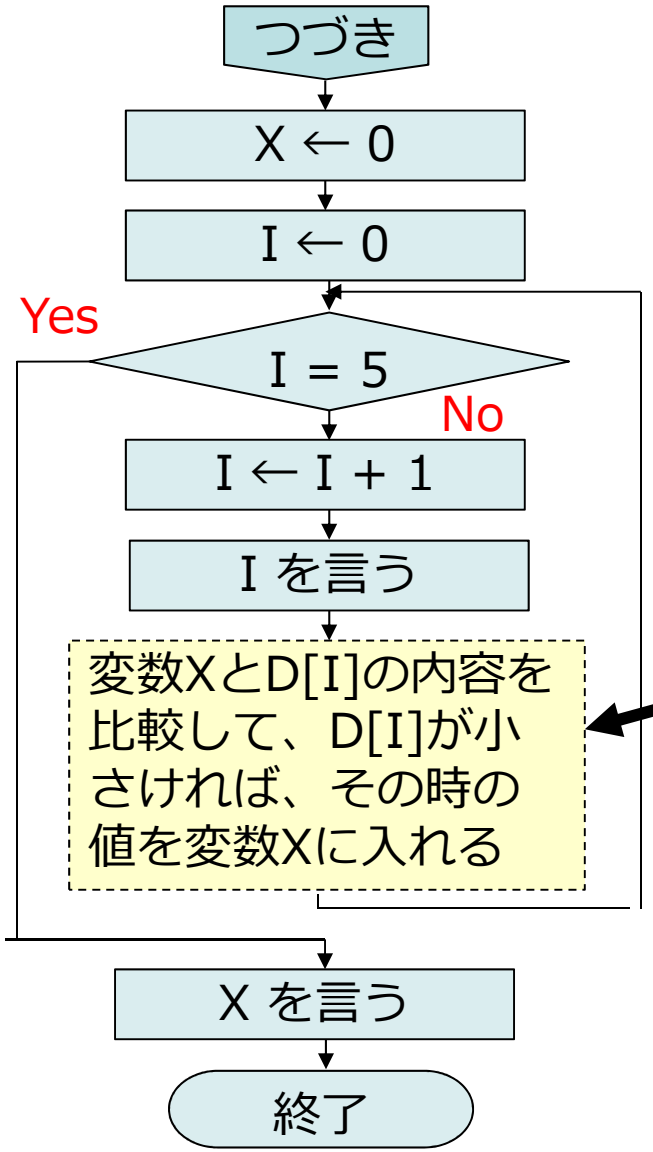
次のスライドにもヒントがあります。

33

課題7補足:「リストの中の一番小さい数を見つける」のヒント



リストDに5個の数を
入力する部分
(乱数を使って入れても
いいです。)

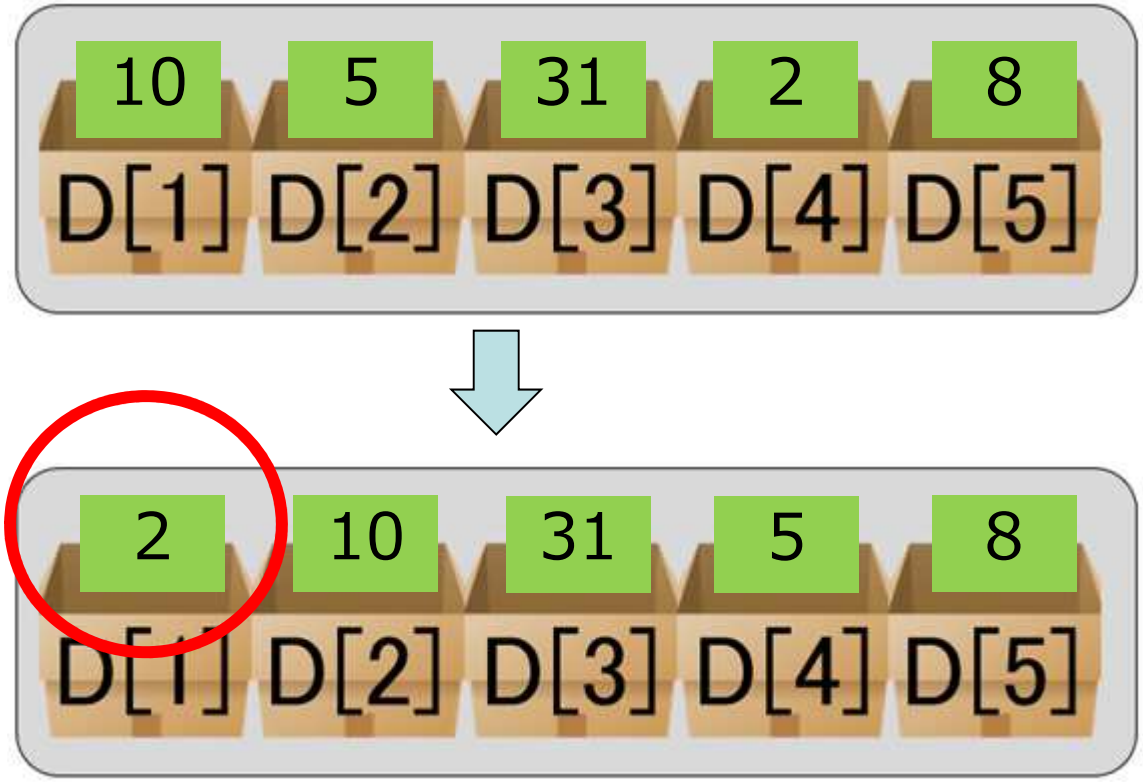


この部分を考えて、プログラムを完成されてください。

課題8: 一番小さい数を配列の先頭に入れ替える

予めリストD[1]からD[5]まで数を入れておきます。その中で一番小さな数を先頭(D[1])に入るように入れ替えます。

開発



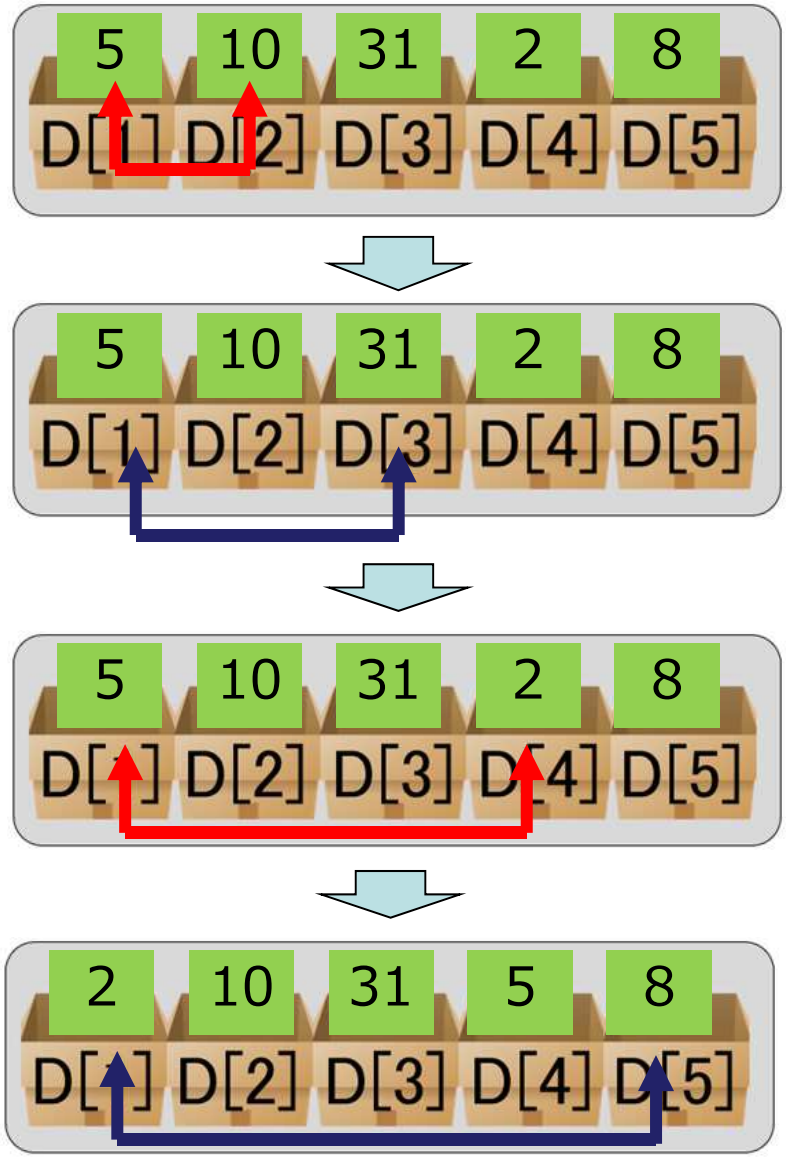
プログラムを実行した後は、

一番小さい数がD[1]に入ります。但し、リストには元のリストにあったすべての数が残っています。

次の3個のスライドにもヒントがあります。

課題8補足: 一番小さい数を配列の先頭に入れ替える

ヒント: プログラムの考え方(実際に紙に数値を書いてやってみよう)



1番目と2番目を比較して5の方が小さいので入れ替え

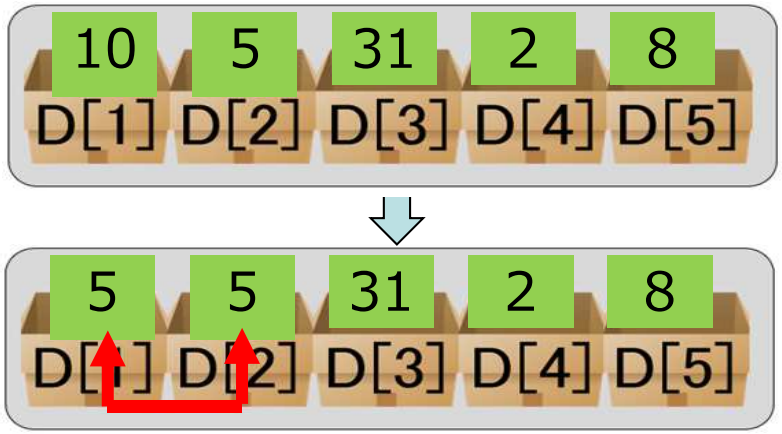
1番目と3番目を比較して5の方が小さいので入れ替えない

1番目と4番目を比較して2の方が小さいので入れ替え

1番目と5番目を比較して2の方が小さいので入れ替えない

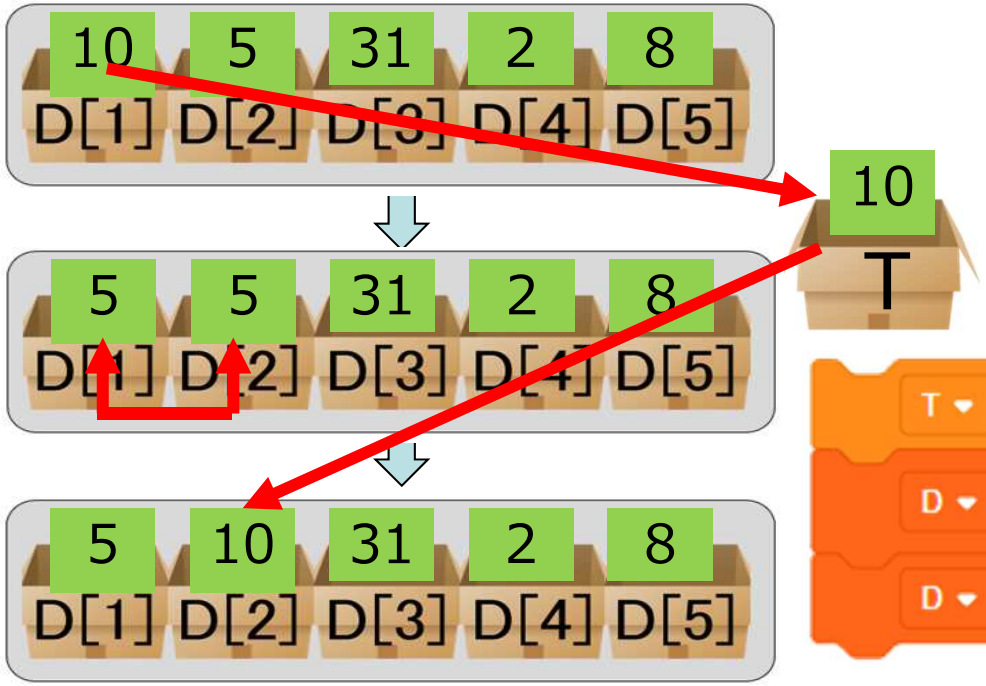
課題8補足: 一番小さい数を配列の先頭に入れ替える

ヒント: 数の入れ変え



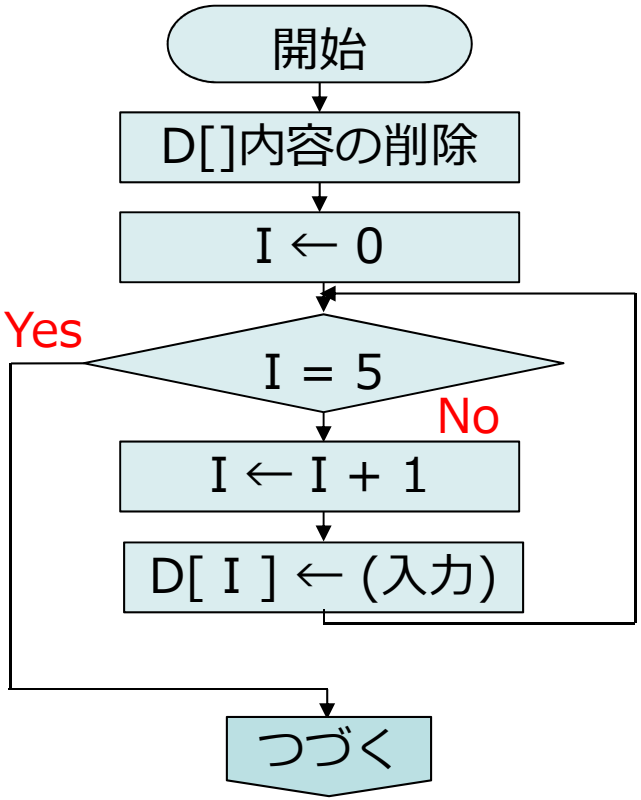
1番目と2番目の入れ替え

いきなり入れ替えると10が残らない

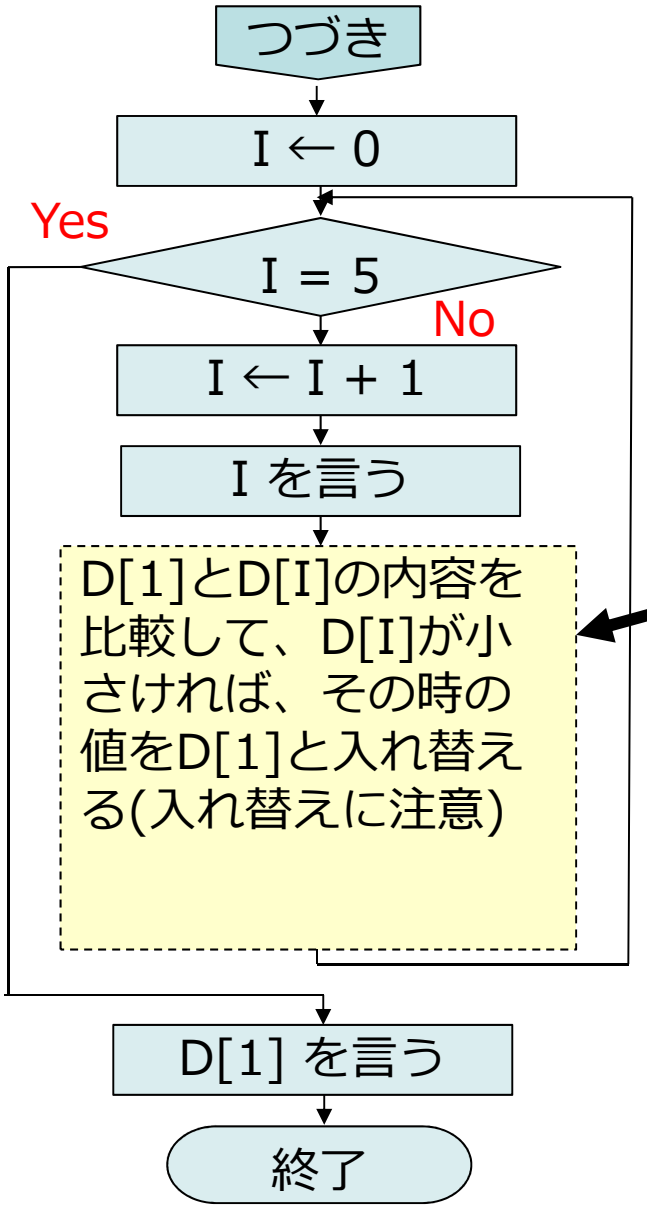


入れ替え先の数を別の変数に保存しておき、それを後で戻す

課題8補足: 一番小さい数を配列の先頭に入れ替える



リストDに5個の数を
入力する部分
(乱数を使って入れても
いいです。)



この部分を考えて、プログラムを完成されてください。

打ち込み:二重繰り返しに挑戦

理解・打ち込み

予めリストD[1]からD[5]まで数を入れておきます。
初めにD[1]からD[5]までの数
次にD[2]からD[5]までの数
その次にD[3]からD[5]までの数
その次の次にD[4]からD[5]までの数
最期にD[5]までの数を言うプログラムを作ります。

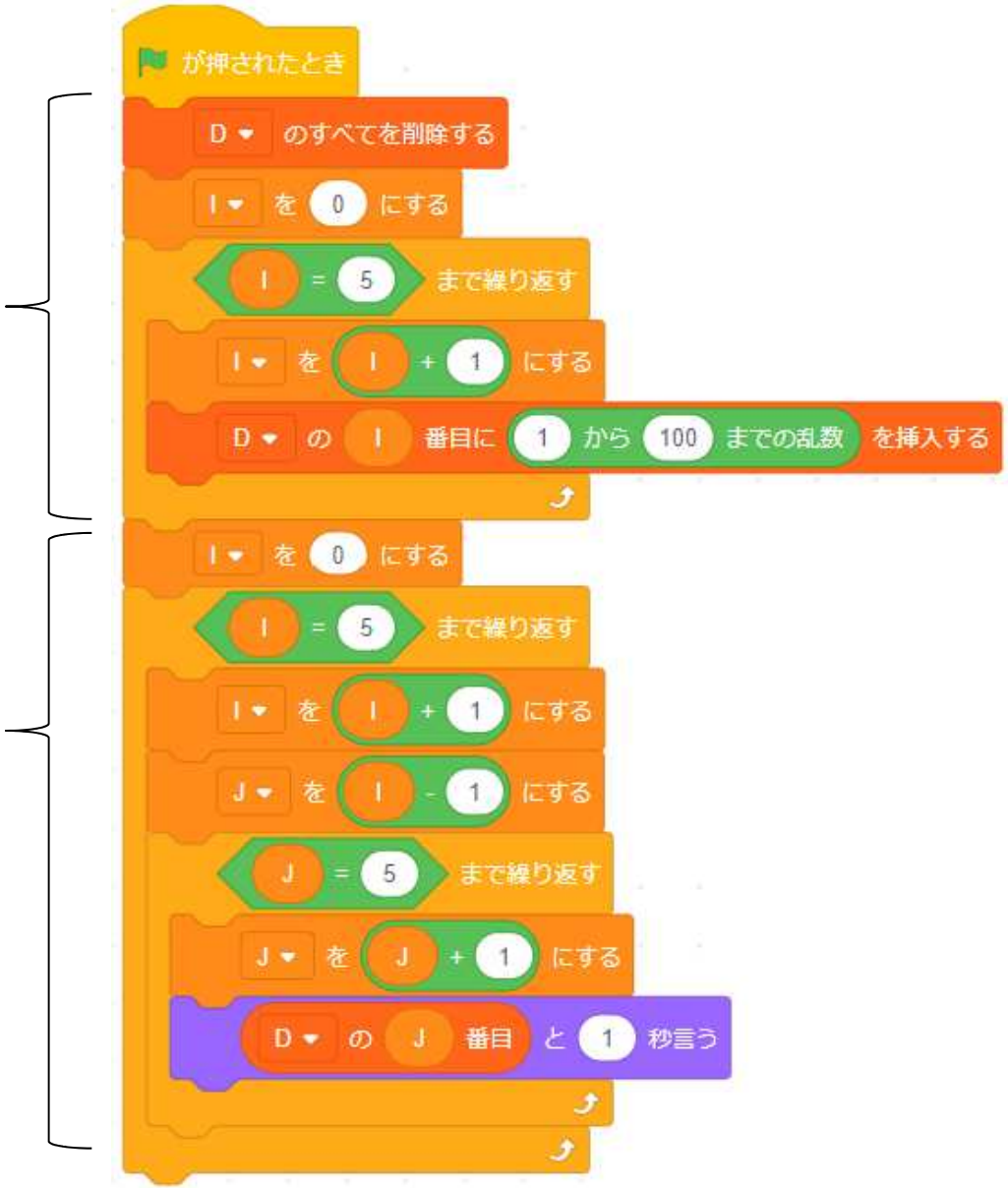
次のスライドにプログラム
次の次のスライドにフローチャート
次の次の次のスライドに解説
があります。

打ち込み:二重繰り返しに挑戦

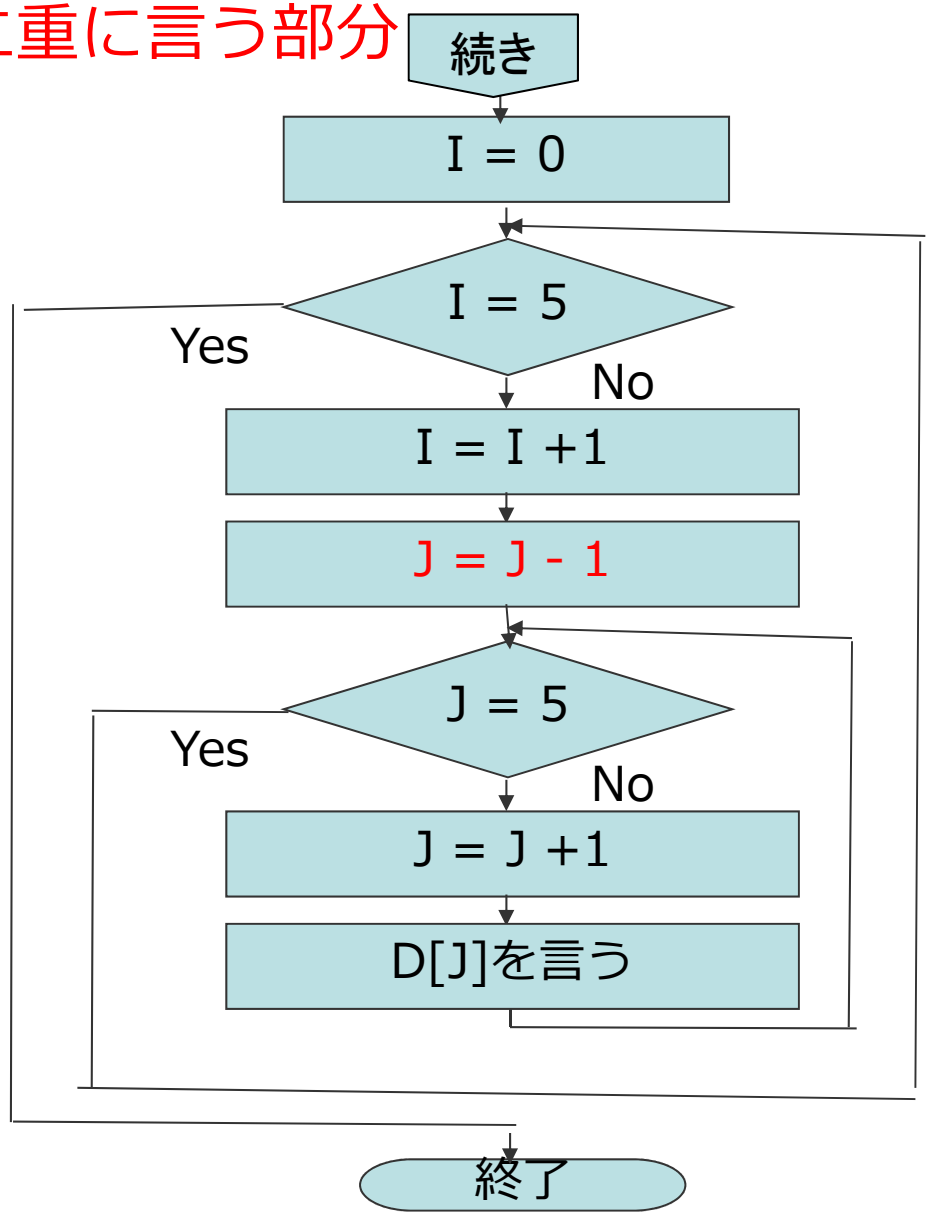
理解・打ち込み

リストに値を設定
している部分
(乱数で設定)

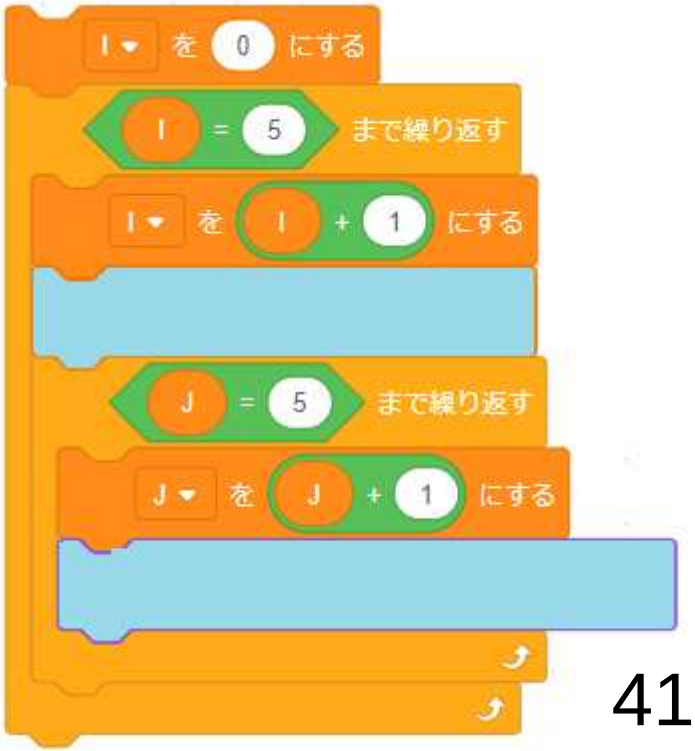
二重に言う部分



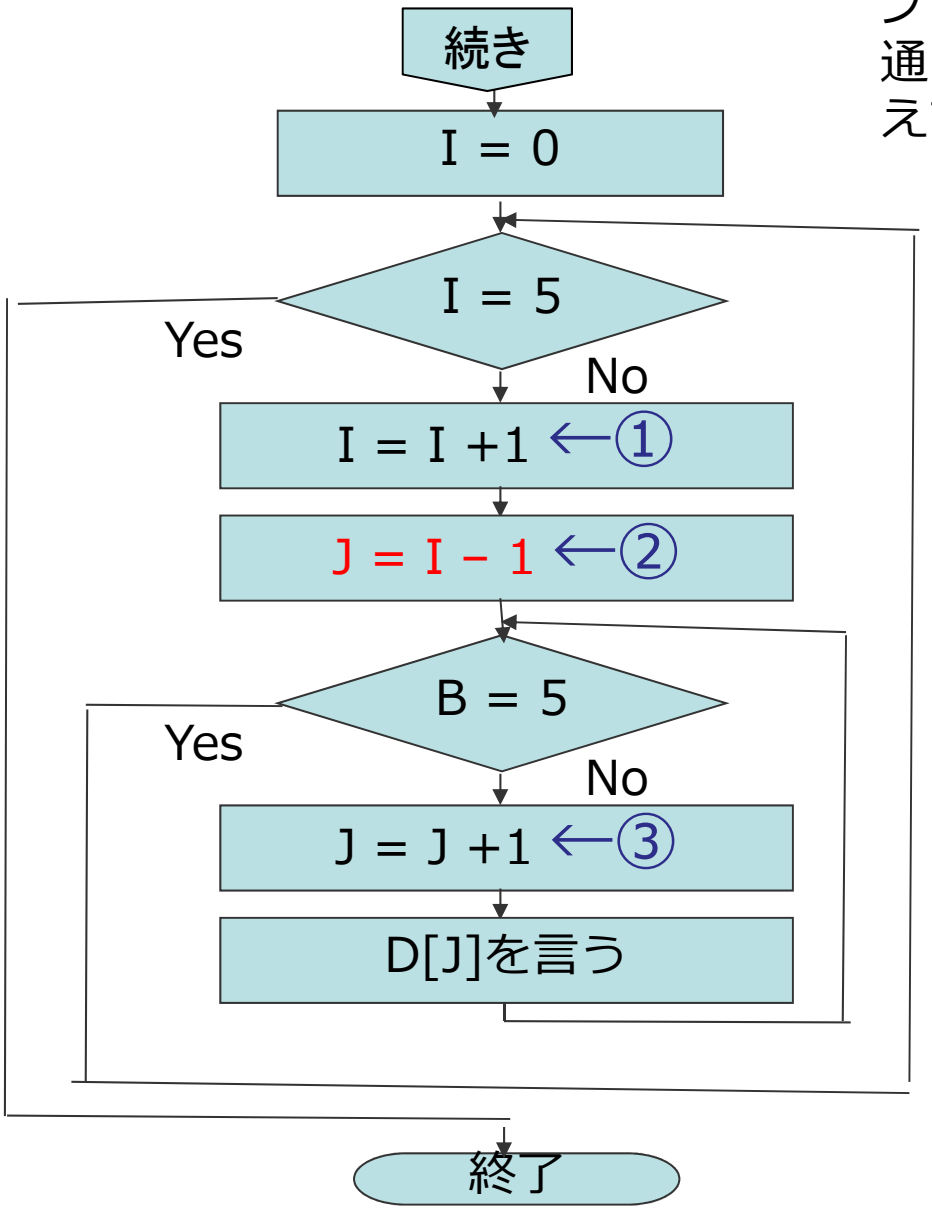
二重に言う部分



フローチャートはごちゃごちゃしていますが、プログラムで見ると、
Iを使った大きな繰り返しの中にJを使った繰り返しが入っている形になります。



二重繰り返しに挑戦



二重に言う部分

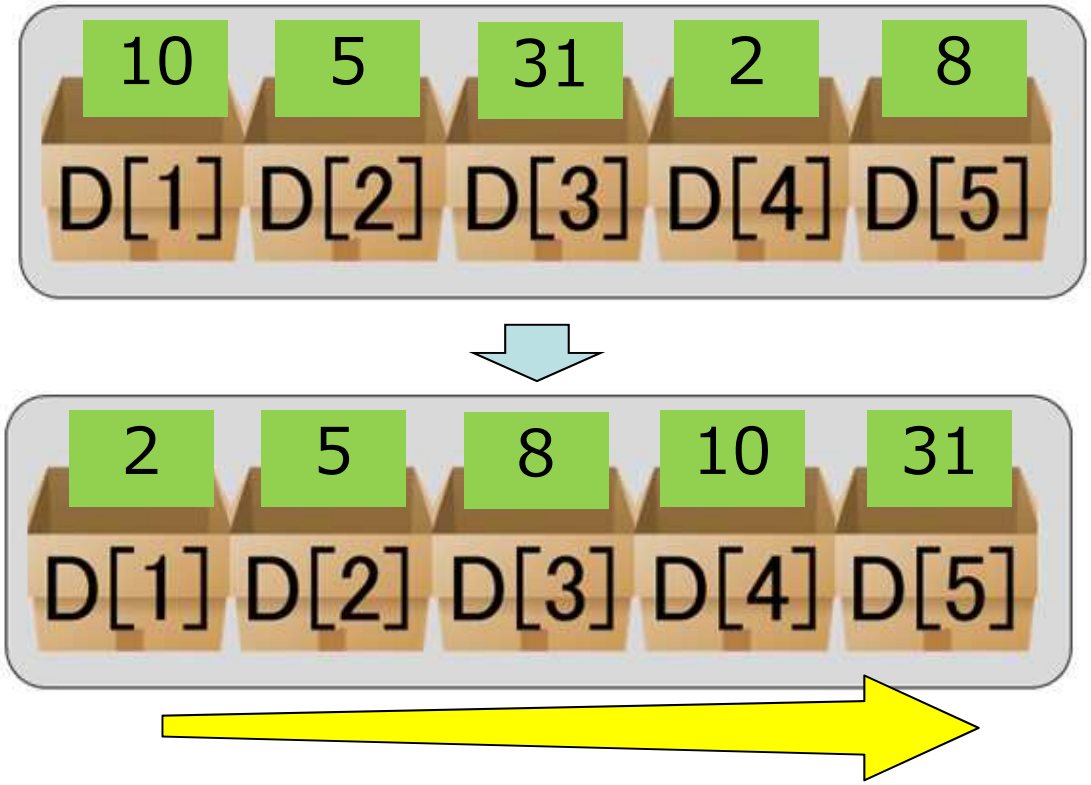
プログラムを動かすと①②③のどこを通って、AとBがどのように変わるか考えてみましょう。

通る場所	I	J
①	1	?
②	1	0
③	1	1
③	1	2
③	1	3
③	1	4
③	1	5
①	2	5
②	2	1
③	2	2
③	2	3
③	2	4
続いていきます		

課題9:数の並び替え

開発

予めリストD[1]からD[5]まで数を入れておきます。この中の数を小さい順番に並び替えてリストD[1]からD[5]に入れなおしてください。



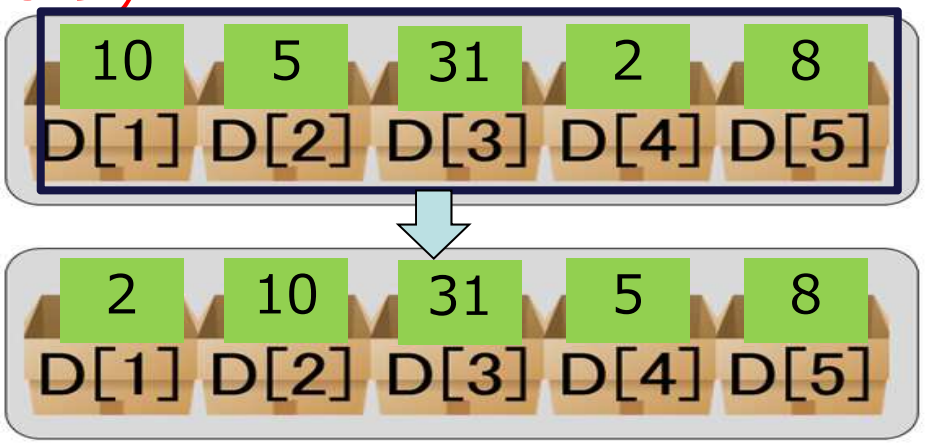
プログラムを実行した後は、数が大きくなるように入れ替えが行われています。

次と次の次のスライドにヒントがあります。

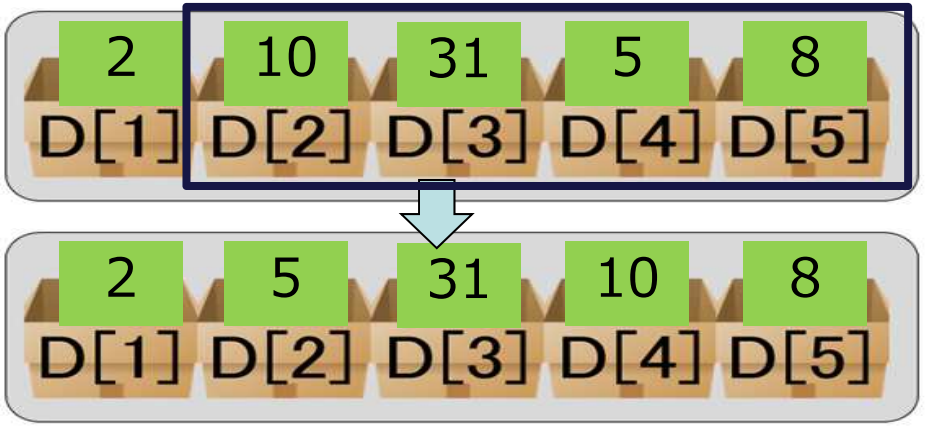
最期のチャレンジ:数の並び替え

ヒント: プログラムの考え方：「一番小さい数をリストの先頭に入れ替える」を繰り返し実施(実際に紙でやってみよう)

1回目
D[1]からD[5]で一番小さい数を配列の先頭D[1]に入れ替える



2回目
D[1]に一番小さい数がはいっているので、D[2]からD[5]で一番小さい数を配列のD[2]に入れ替える



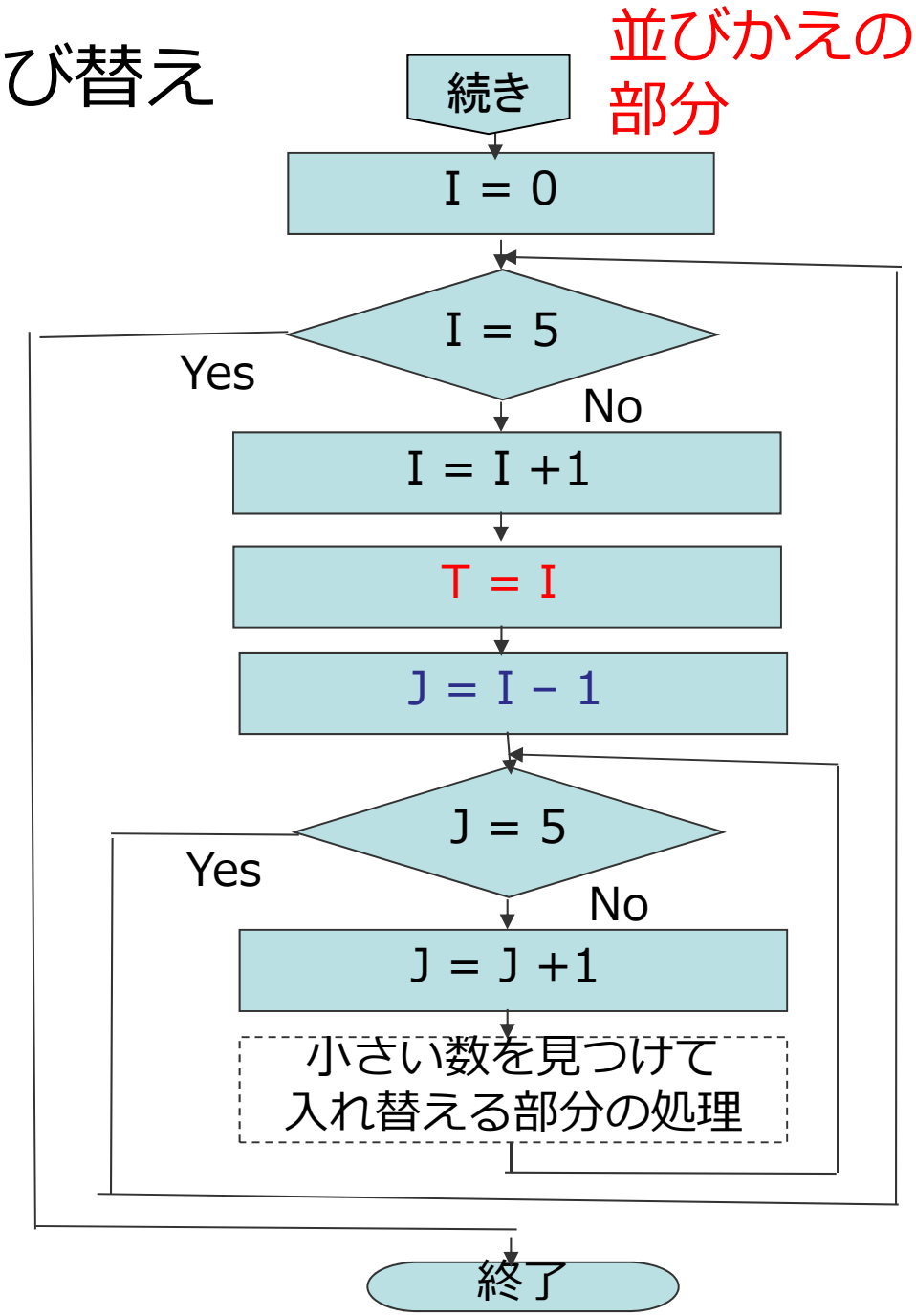
3回目
D[3]からD[5]で一番小さい数を配列のD[3]に入れ替える

3回目
D[4]からD[5]で一番小さい数を配列のD[4]に入れ替える

最期のチャレンジ:数の並び替え

プログラムは①「二重繰り返しに挑戦」の中に、②「一番小さい数をリストの先頭に入れ替える」を組み込んだものになります。

ポイント:
②のプログラムの場合は、一番小さな数が入る配列はD[1]に固定されていました。並べ替えの場合は、これが変わっていきます。右のフローチャートでTを追加したので、これを利用して開発してみてください。



発展課題1: 他の並び替えの方法

開発

今回作成したプログラムは選択ソートという方法を使ったもので、データを並び替える方法はいろいろあります。

次の二つの並び替え方法の考え方やプログラムをWebで調べて作成してください。

バブルソート

クイックソート

発展課題2:並び替えの方法の処理速度の違い

開発

最期の課題です。

データの並び替えに、いろいろな方法があるのは、用途や処理速度に違いがあるからです。

今回開発した
選択ソート/バブルソート/クイックソート
で処理速度がどのように違うか測定してみましょう。

ここに並び替えをするプログラムを入れます

Scratchでタイマーを使うと処理にかかった時間が測定できます。