

# 情報の授業

## アルゴリズムとプログラム Ver. 4.

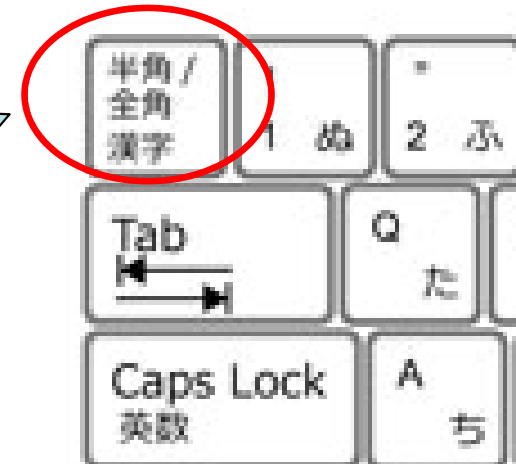
### 学習教材

半角の数字 1, 2, 3

漢字(全角)の数字 1, 2, 3

Scratchでは、漢字の数字を入れると正しく動作しません。

今回の実習中は、漢字(日本語)の入力モードを切って(オフ)使ってください。



学習教材をスマホで表示しながら、やりたい人はここから



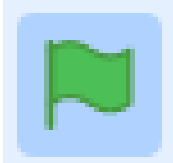
# 準備運動1: Scratchの四則演算

理解・打込み



そのまま打ち込む  
プログラム

プログラムが出来たら旗を  
押して動かしてみよう



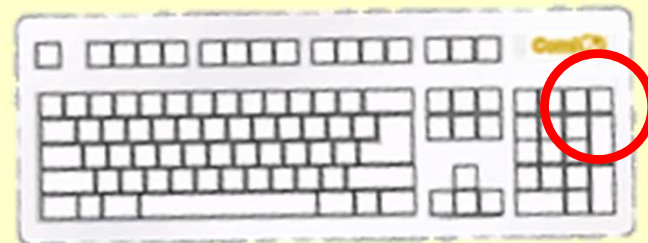
まず、準備運動で、四則演算(足し算、引き算、掛け算、割り算)のプログラムを作って動かしてみます。

- ・ 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。

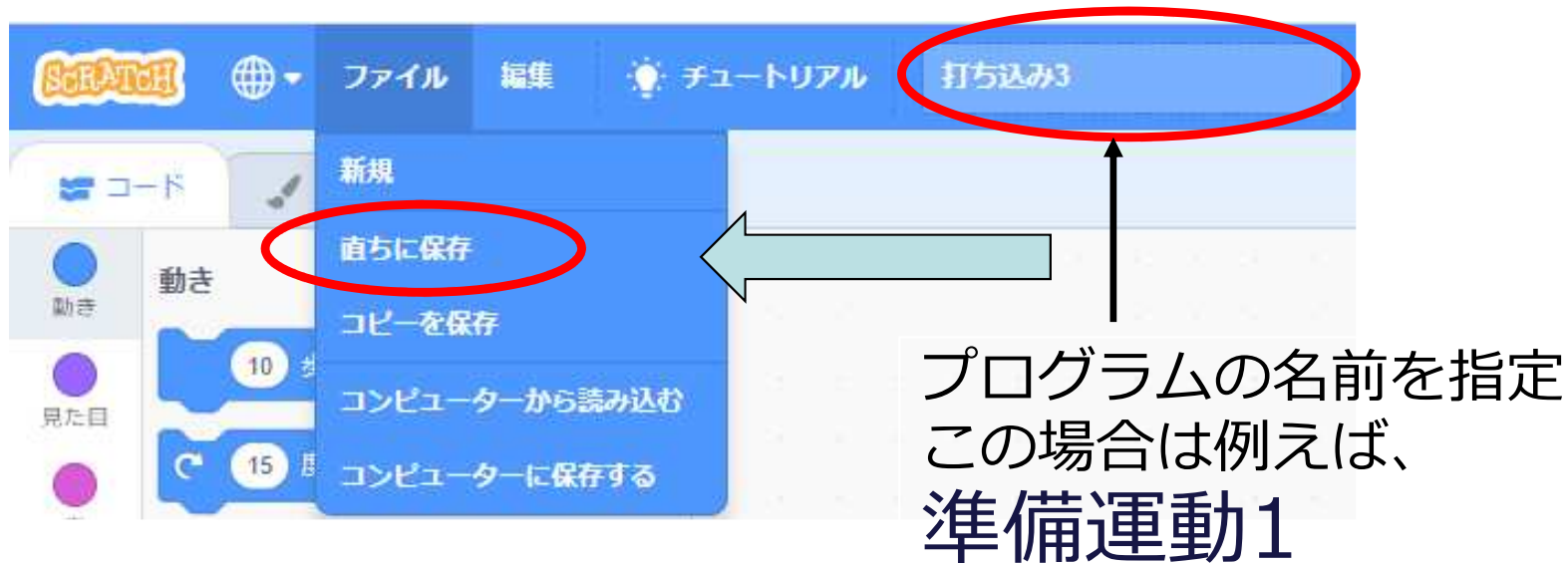
掛け算 \*

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)



# プログラムに名前をつけて保存

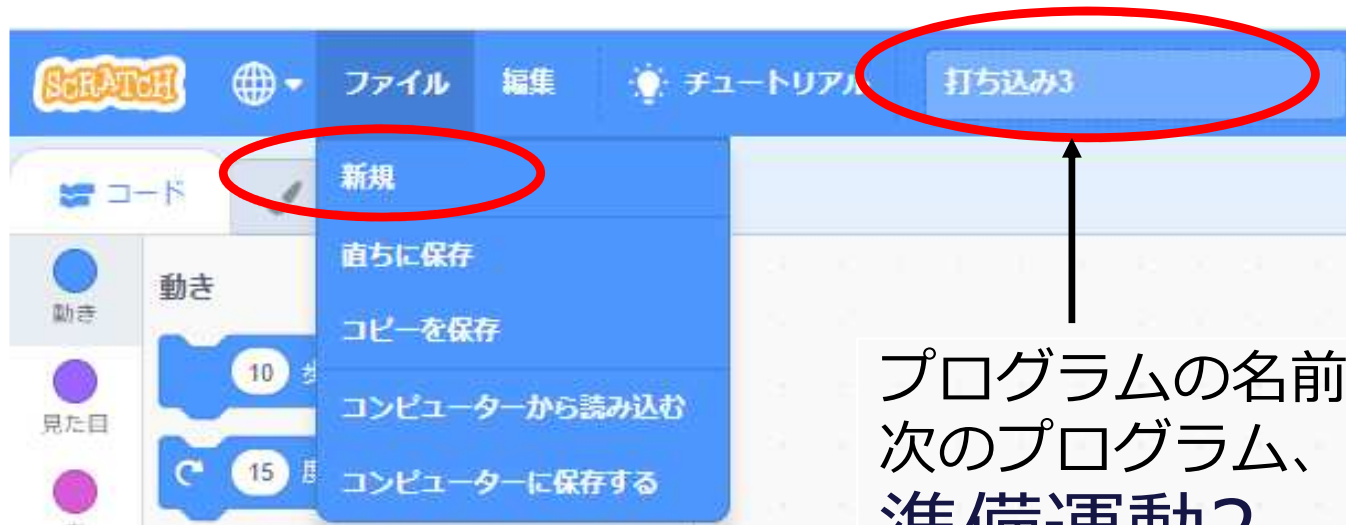


これからの課題や打ち込むプログラムは前に作ったものを改造すると楽なものが多くなります。個々のプログラムにつけて保存したり、コピーを保存で流用したりしましょう。

今作っているプログラムはネット上に自動保存されます。確実に保存したい人は**直ちに保存**を指定します。

# 新しいプログラムを作る

理解



プログラムの名前を指定  
次のプログラム、  
**準備運動2**

新しいプログラムは  
**新規**で作成

## 準備運動2: 円から韓国ウォンに変換。

理解・打込み



そのまま打ち込む  
プログラム

変数の **A** を使う



次のスライド  
のヒント  
も見て

日本円の金額を入れて、韓国のウォンがいくらになるか計算します。  
1円=11ウォンで計算しています。

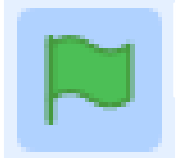
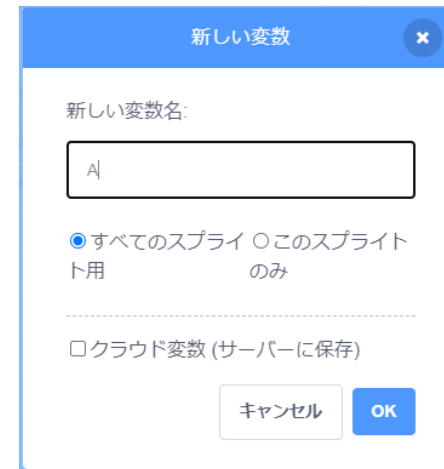
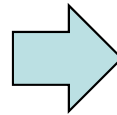
# 準備運動2: 円から韓国ウォンに変換。

ヒント

Aという名前の変数を作って使います。



[変数を作る]



プログラムを動かす。

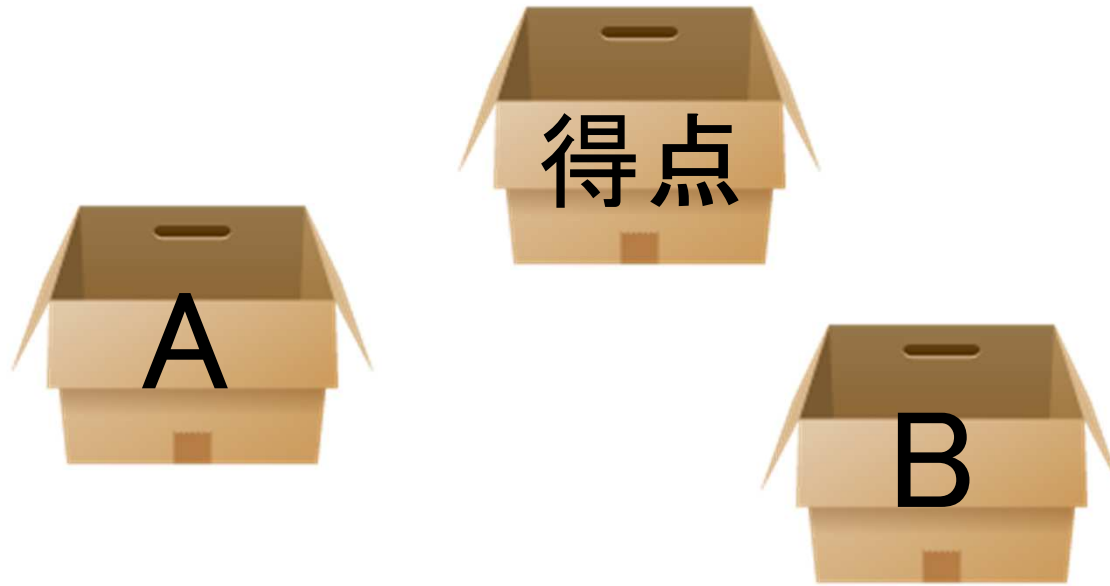


キーボードから入力。



入力した数字が入る。

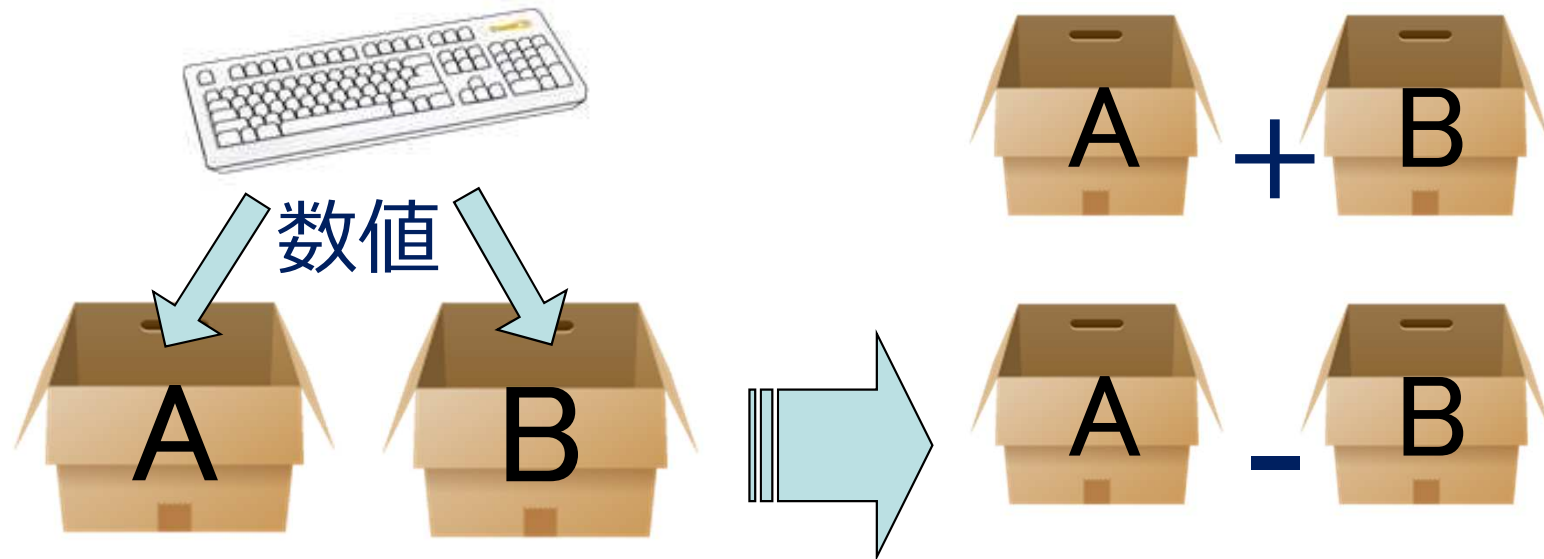
# プログラムと変数



変数は数が入る箱みたいなものです。

# 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。

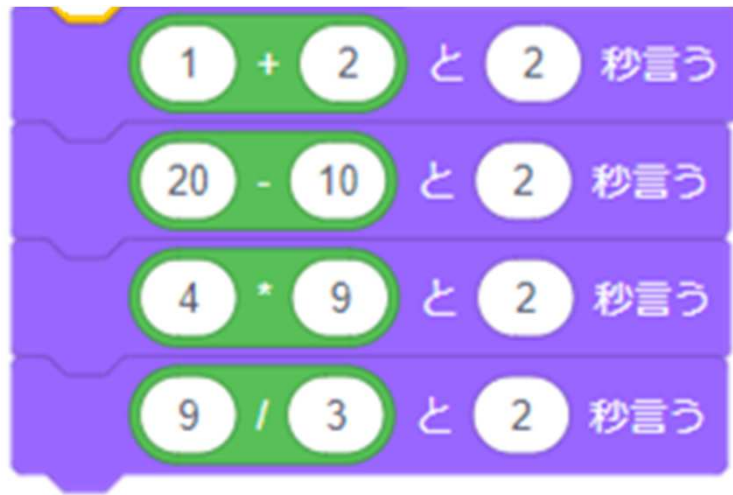
部品  
01

部品  
02

次のスライドの  
ヒントも見て

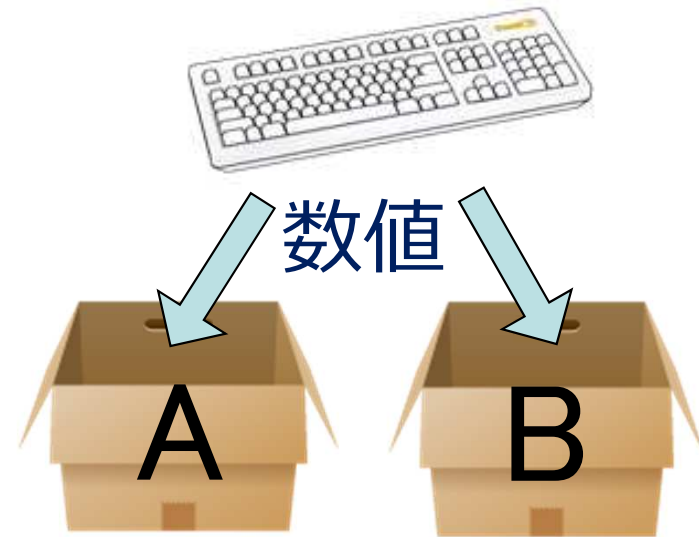
# 課題1: 入力した2個の数で四則演算

ヒント



足し算、引き算、掛け算、割り算は準備運動1でやっています。

直接数字を指定して入れけど、これをAとBの変数にする。



部品  
02

変数Aと変数Bに入力する方法は、部品カード02を見てください。

# 打ち込み1: 合格判断

理解・打ち込み



そのまま打ち込む  
プログラム

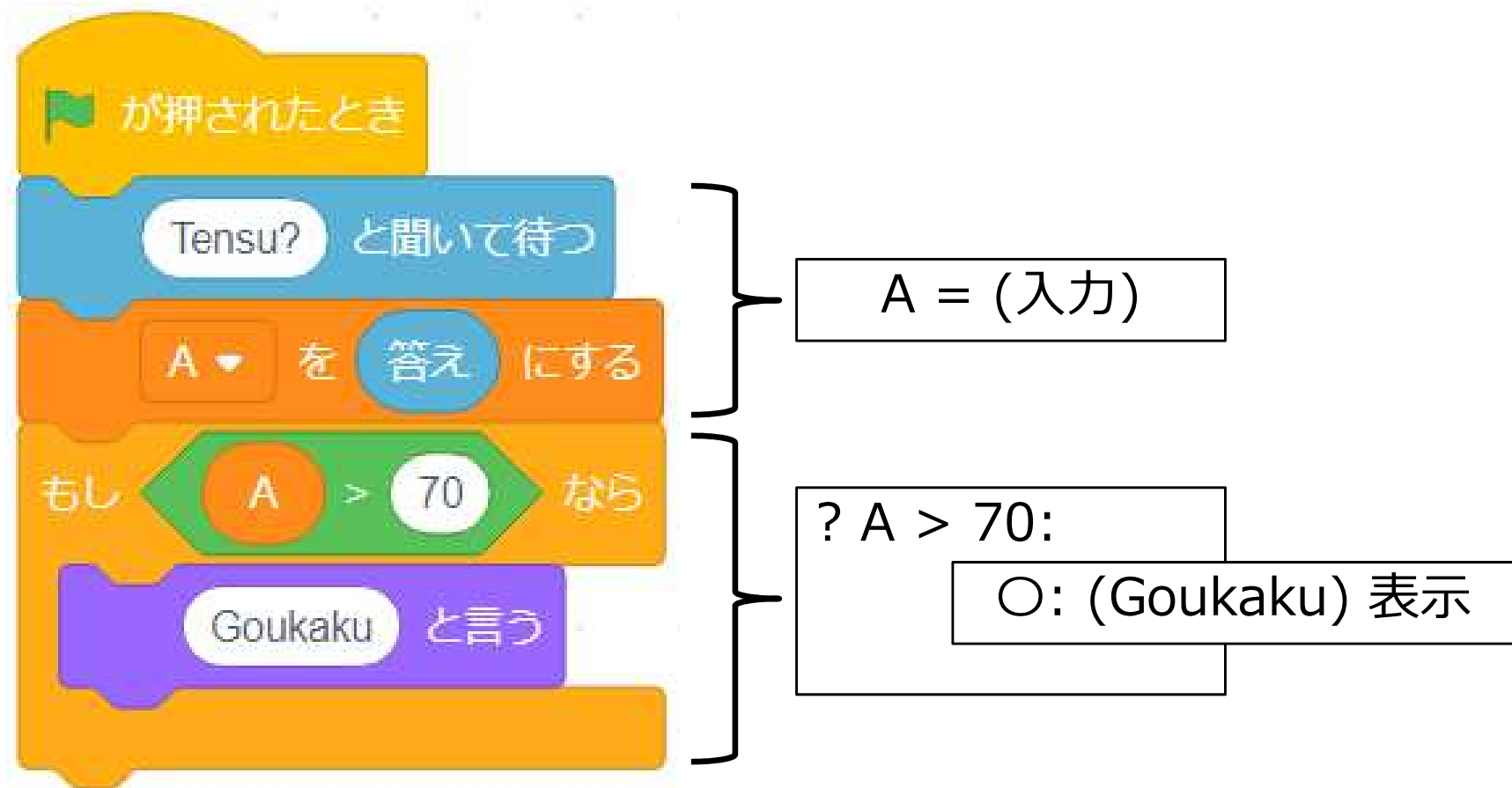


変数Aに数を入力して、70より大きければ(70は入らない)、ネコにGoukaku(合格)と言わせてみます。

次のスライドの  
理解も見て

# 打ち込み1: 合格判断の図式

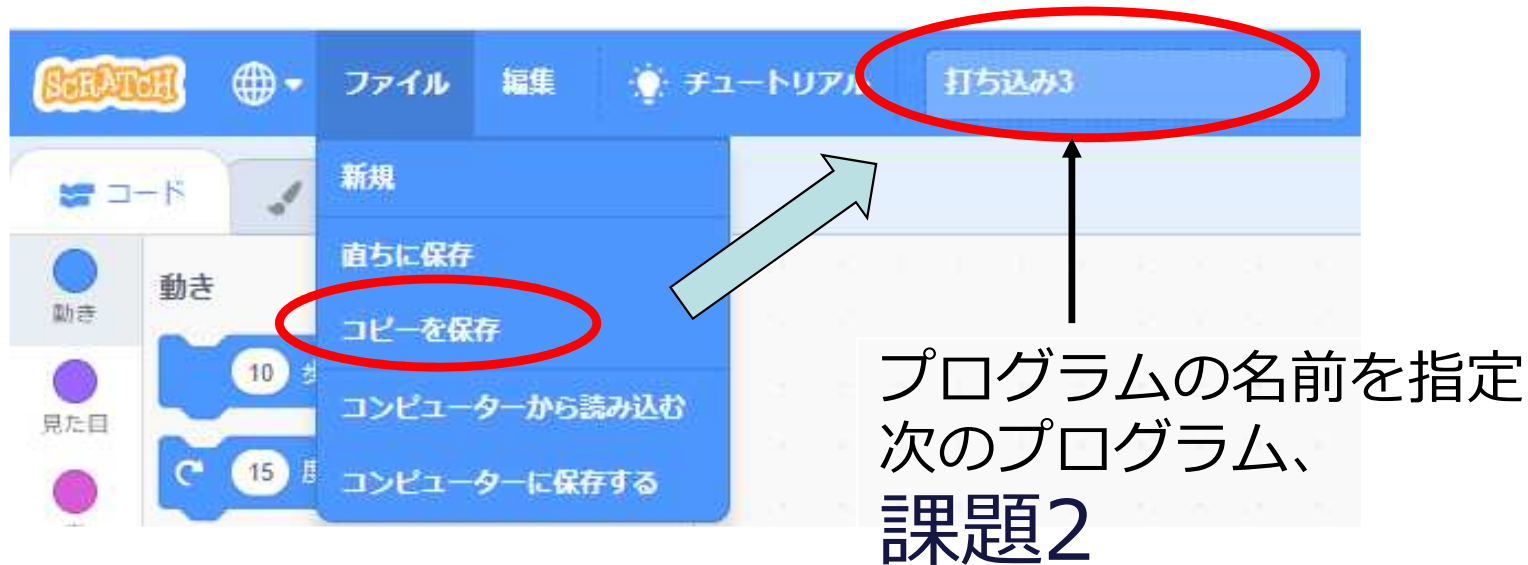
理解



部品  
03

部品カード03も見てく  
ださい。

# 他のプログラムを流用して作る



良くあるプログラム  
たぶん

打ち込み1

を利用して新しいプログラムを新規につくる場合は  
コピーを保存で作成

## 課題2: 合格不合格判断

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**Goukaku**(合格)と言わせます。  
そうで**なければ**、ネコに**Fugoukaku**(不合格)と言わせます。

部品  
04

ヒント: **部品04**を参考にして作ってみてください。



## 課題3: 成績A～C

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)、**Seiseki A**(成績A)、  
50より大きい**なら**(50は入らない)、**Seiseki B**(成績B)、  
それ以外(50以下だったら)、**Seiseki C**(成績C)、  
とネコに言わせます。



ヒント: 課題2を流用して  
(コピーして保存)して作る。

部品  
05

ヒント: **部品05**を参考にし  
て作ってみてください。

# 打ち込み2: 1から10までの数を言う

理解・打込み



1,2,3,・・・,10までの数を  
ネコと順番にと言わせてみます。

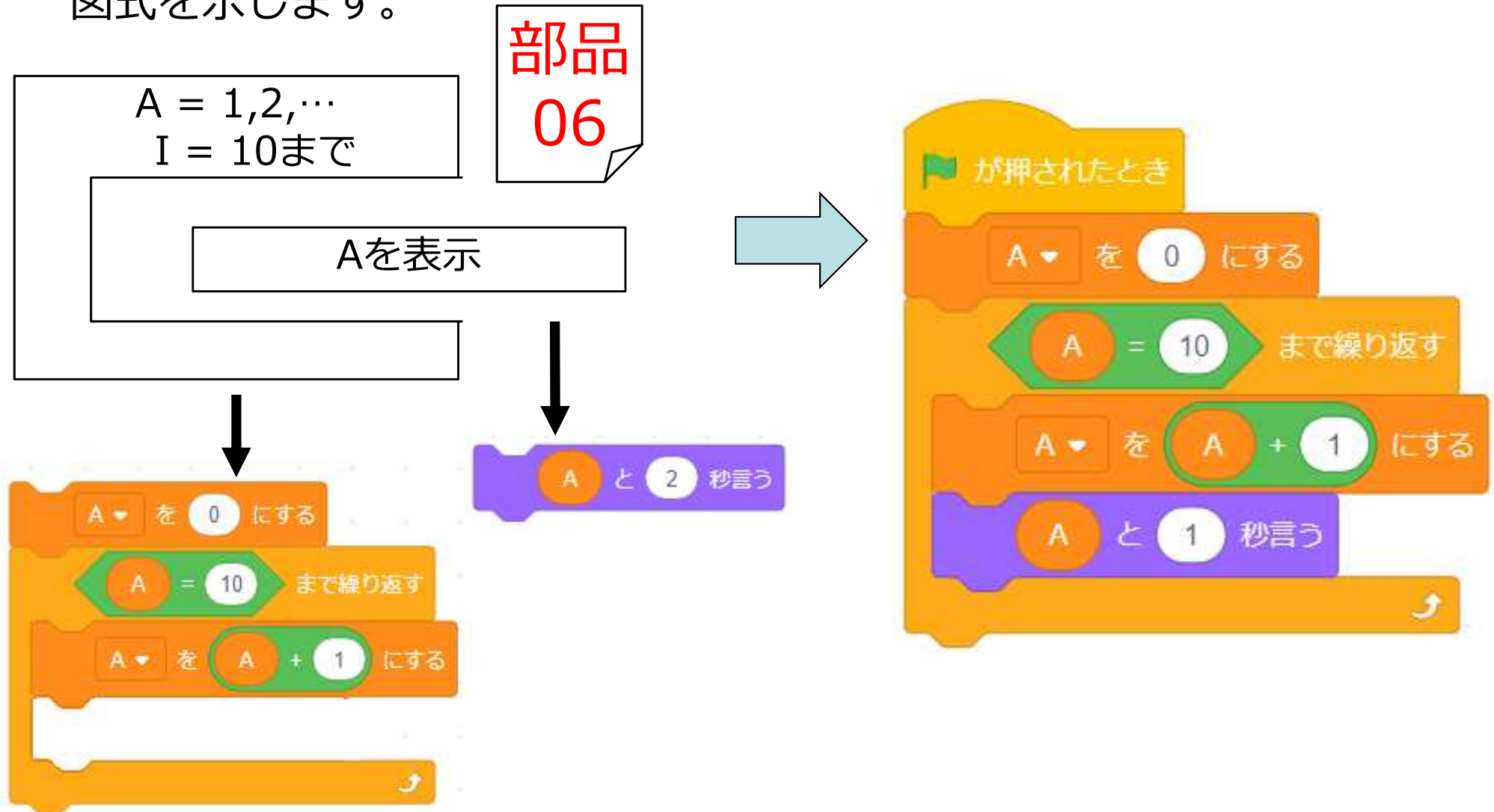
部品  
06

そのまま打ち込む  
プログラム

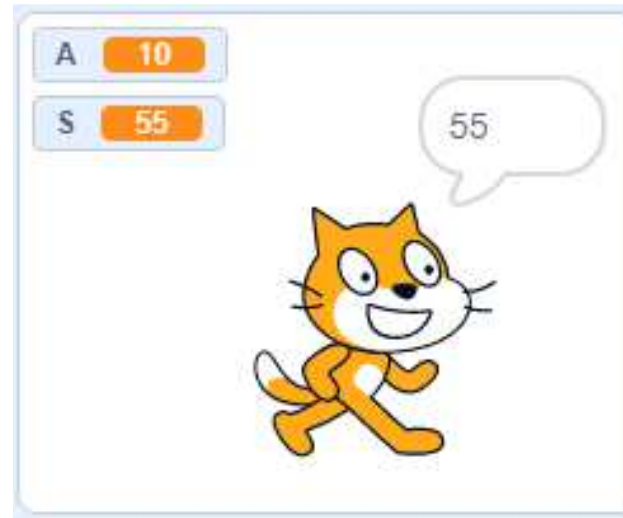
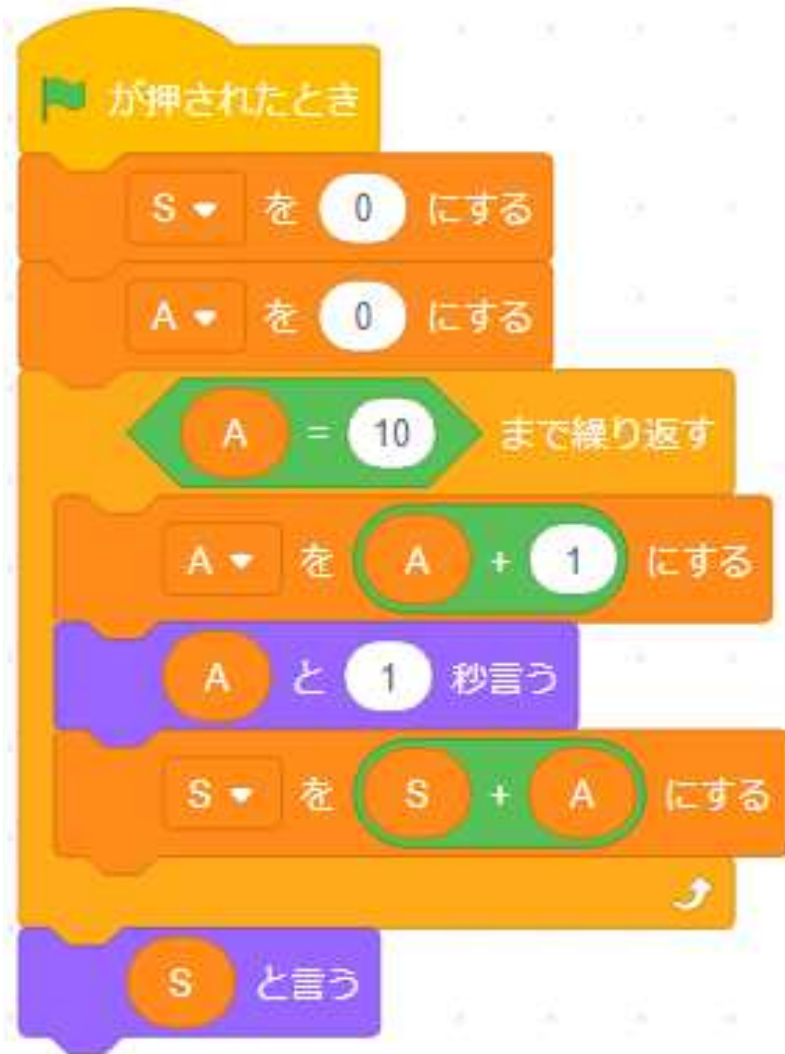
次のスライドの  
理解も見て

# プログラムと図式

プログラムもだんだん複雑になってきます。プログラムがどのような部品でくみあがっているか分かりやすくするため、今後ヒントで図式を示します。



## 打ち込み3: 1から10までの合計



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。

ヒント: **打ち込み2**を流用して(コピーして保存)して作る。

そのまま打ち込む  
プログラム

次のスライドの  
理解も見て

# プログラムの図式

「1から10までの合計」は次のような図式になります。

$S = 0$

$A = 1, 2, \dots$   
 $I = 10$ まで

Aを表示

$S = S + A$

Sを表示

部品  
06

部品6を見てね。



## 課題4: 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1:

打ち込み3: 「1から10までの合計」のプログラムを流用して(コピーして保存)して作る。

### ヒント2: 次のスライド

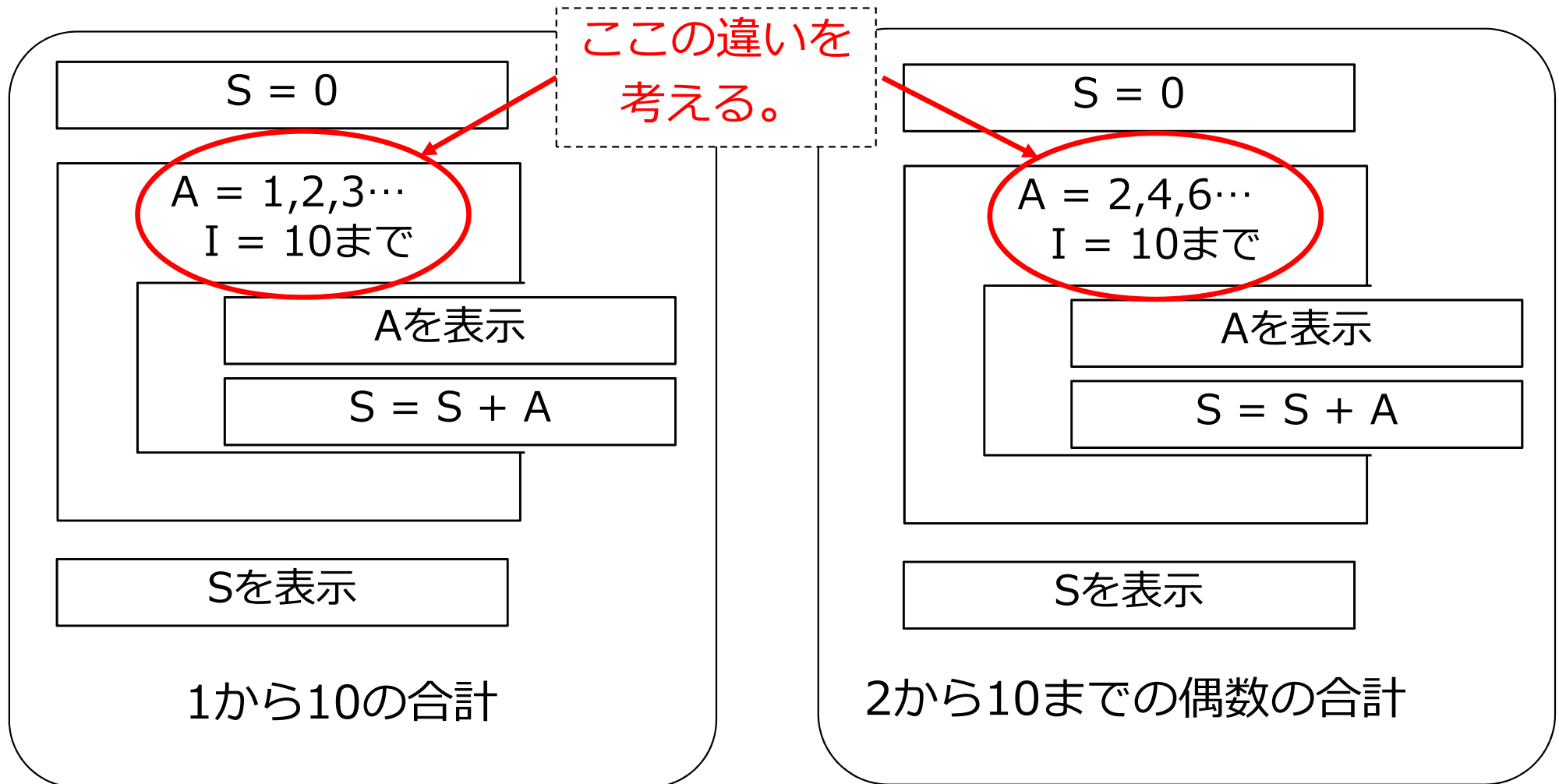
まず、2から10までの偶数の合計を求めるプログラムを作ります。

### ヒント3: 次の次のスライド

次に、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。

## 課題4 ヒント2: 2から10以下の偶数の合計

ヒント



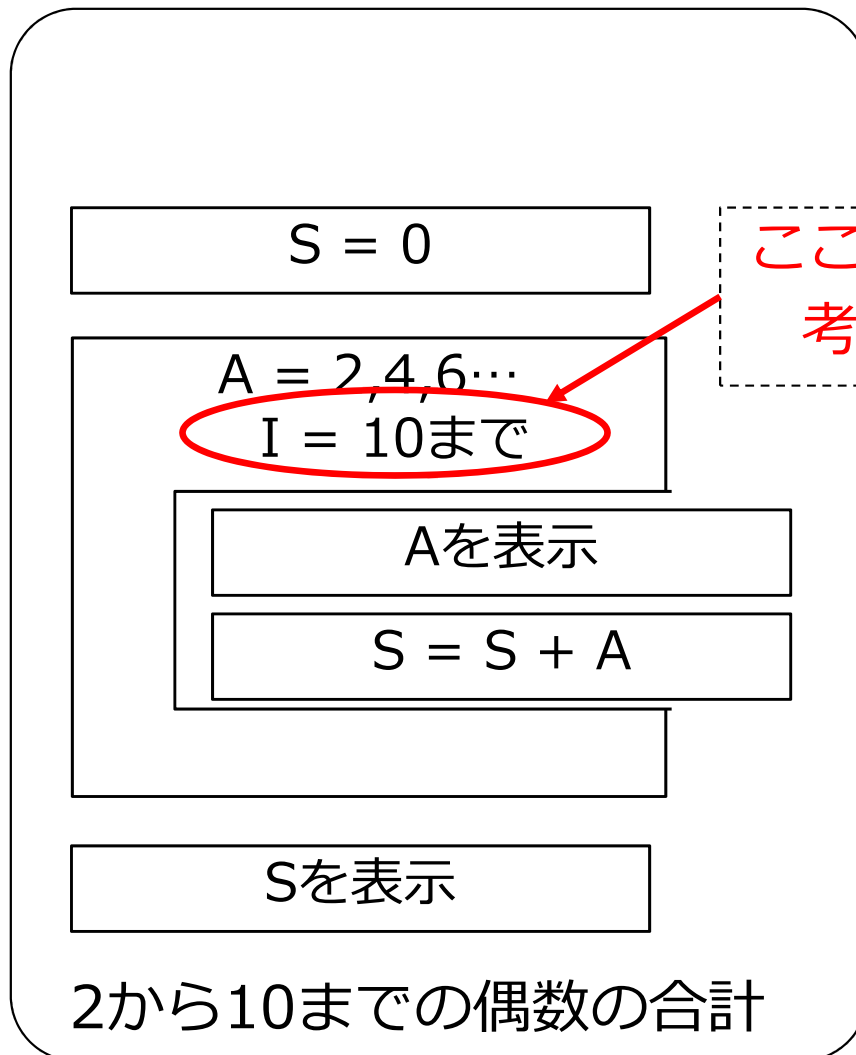
部品  
06

2, 4, 6となるように部品06を少し  
変更して使います。

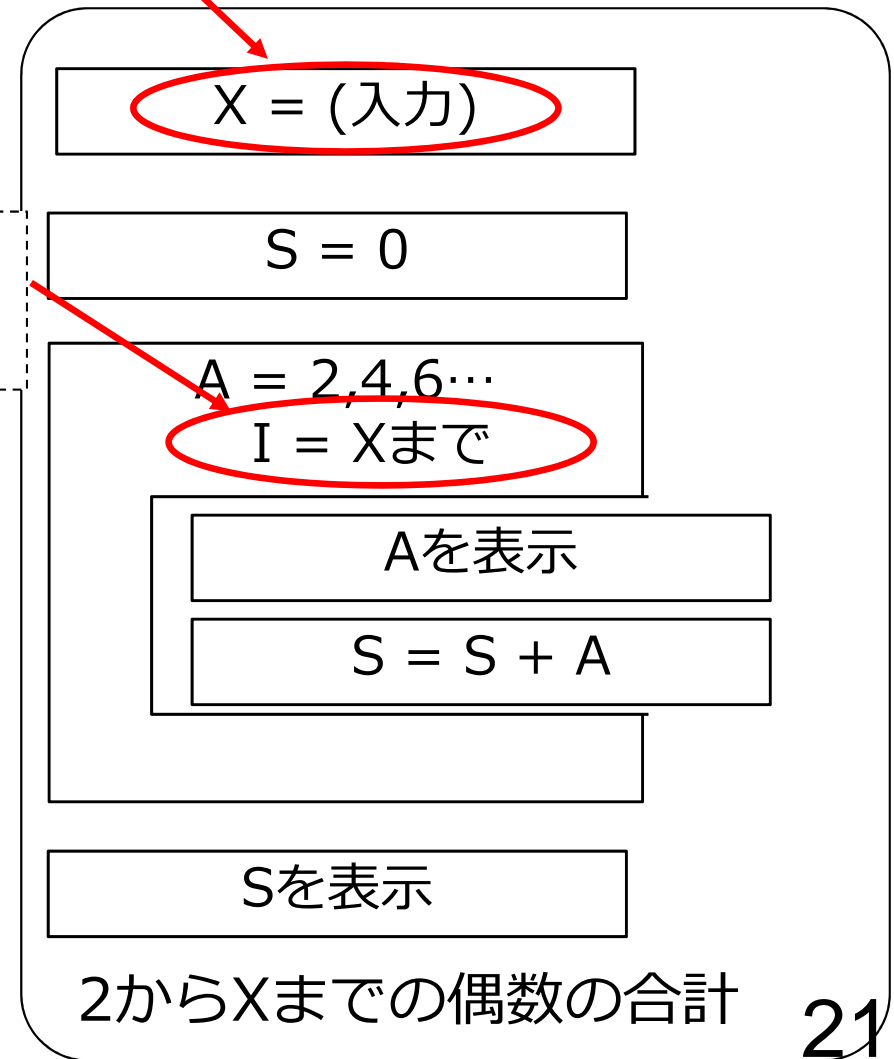
# 課題4補足: 2からX以下の偶数の合計

ヒント

部品  
02



この違いを  
考える。



# 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、 $A, B, C, D, E \dots$ 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: 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。



そのまま打ち込む  
プログラム



ネコをクリックするとDai-Kichi(大吉), Chu-Kichi(中吉), Syo-Kichi(小吉), Kyou(凶), Dai-Kyou(大凶)とランダムに言います。

部品  
07

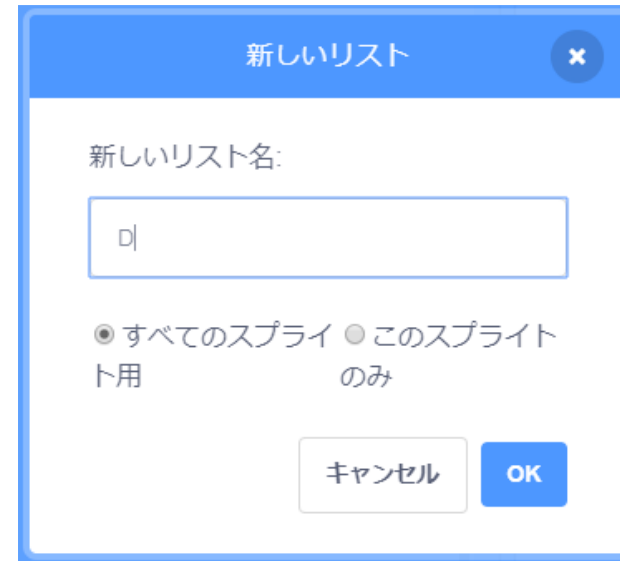
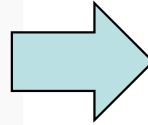
次, 次の次、次の次の次の  
スライドに作り方のヒントも見て

## 打ち込み4: 配列(リスト)を作る

ヒント



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。 24

# 打ち込み4:配列(リスト)に値を入れる

ヒント

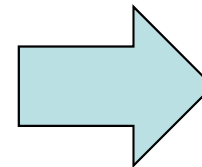
D	
1	Dai-Kichi
2	Chu-Kichi
3	Syo-Kichi
4	Kyou
5	Dai-Kyou

+ 長さ 5 =



D	
(空)	

+ 長さ 0 =



D	
1	Dai- X

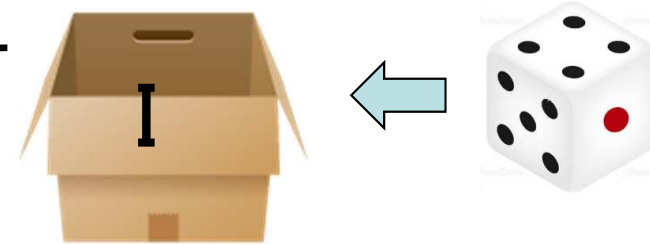
+ 長さ 1 =

+ を押す

入力する

## 打ち込み4: プログラムの動作

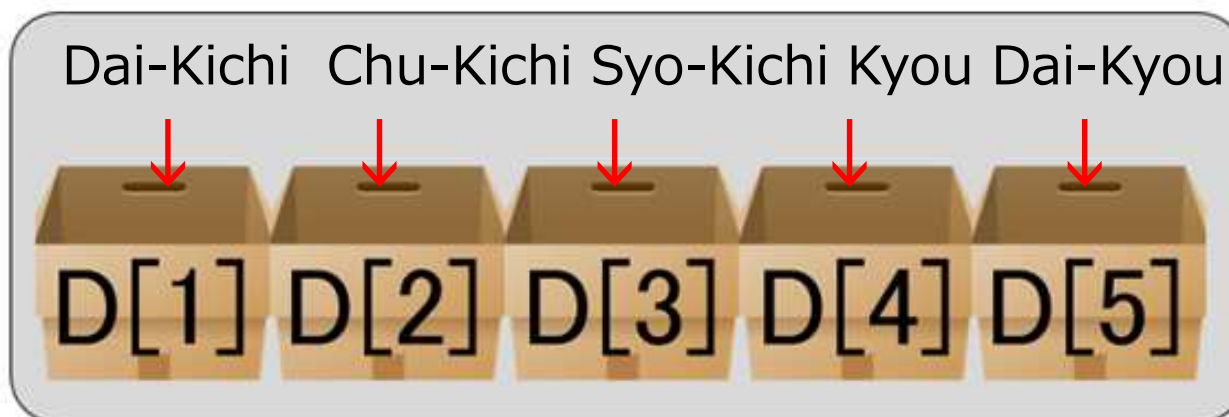
ヒント



変数I(アイ)に1~5の  
どれかの数が入る。

ヒント: 変数 I(アイ)を  
作って使います

D[ I(アイ) ]番目 の内容を言  
う(Iに入っている数に対応し  
たD[]の内容)



# 打ち込み5:リストを使った5つの数の合計

理解・打込み

D	
1	6
2	10
3	8
4	3
5	6
+ 長さ 5 =	



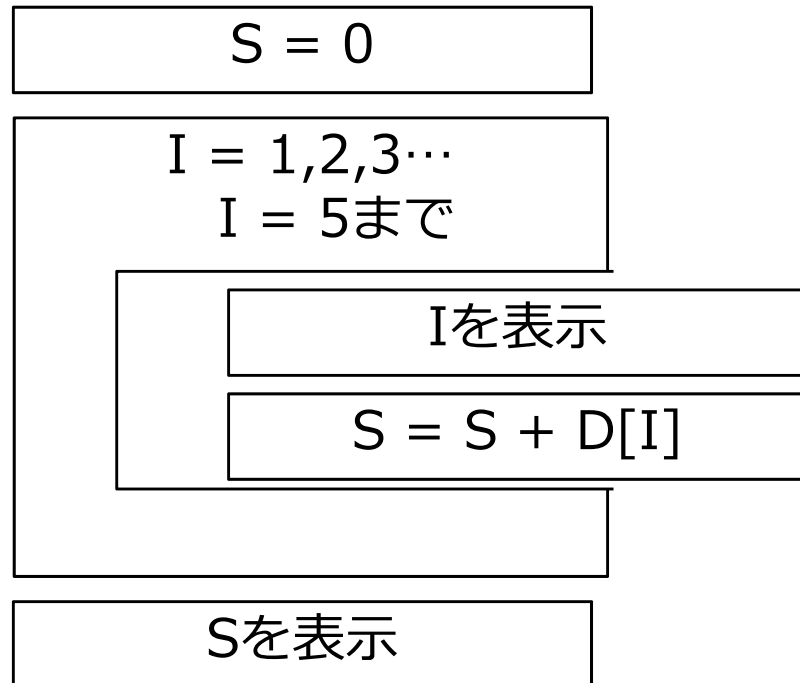
そのまま打ち込む  
プログラム

いきなりプログラムが難しくなりました。あらかじめリストDに入れておいた5個の数の合計を計算します。

次のスライドの  
理解も見て

# 打ち込み5補足:リストを使った5つの数の合計

理解



リストDに5個の数を変数にSに加えていって合計を計算

部品  
07

部品  
09

## 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。

- ・ 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
- ・ 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



# 変数/リストシートを使って考えよう

理解

アルゴリズムとプログラム実習用シート

○変数



○リスト(配列)

D[ ]



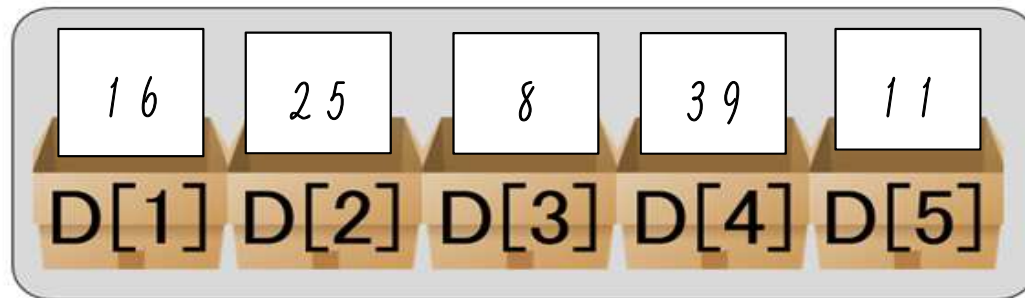
段ボールの箱を変数として見立て、変数/リストシートが用意されています。

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、紙に書いてみましょう。

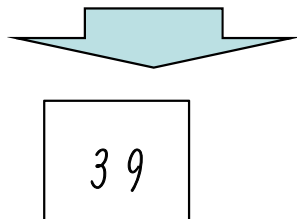
## 課題5: リストの中から数を探す(検索)

### 課題

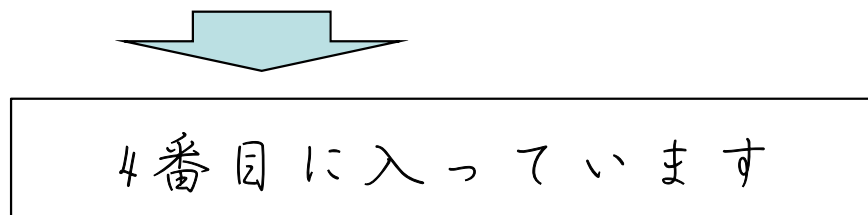
検索というこんな動作をするプログラムを作ります



初めにリストに数を入れておきます



次に探す数を決めます



**ヒント:**

打ち込み5:リストを使った5つの数の合計」のプログラムを流用して(コピーして保存)して作る。

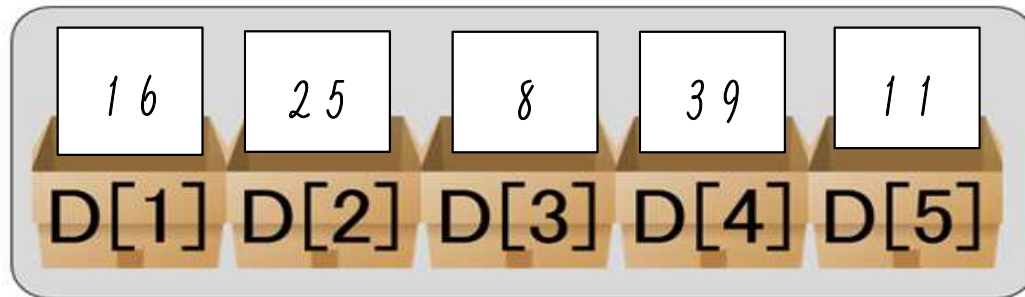
次のヒント1~3のスライドも見て

プログラムで、探す数がリストDの何番目に入っているか見つけ出します。

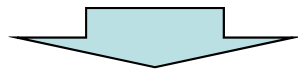
## 課題5: リストの中から数を探す(検索)

ヒント1

プログラムの中で次のように変数使います

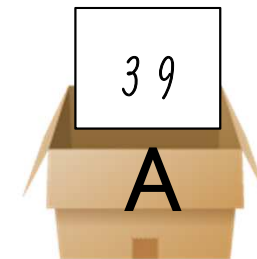


初めにリストに数を入れておきます

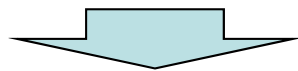


39

A = (入力)



次に探す数を決めます(探す数を変数Aに入力します)



4番目に入っています



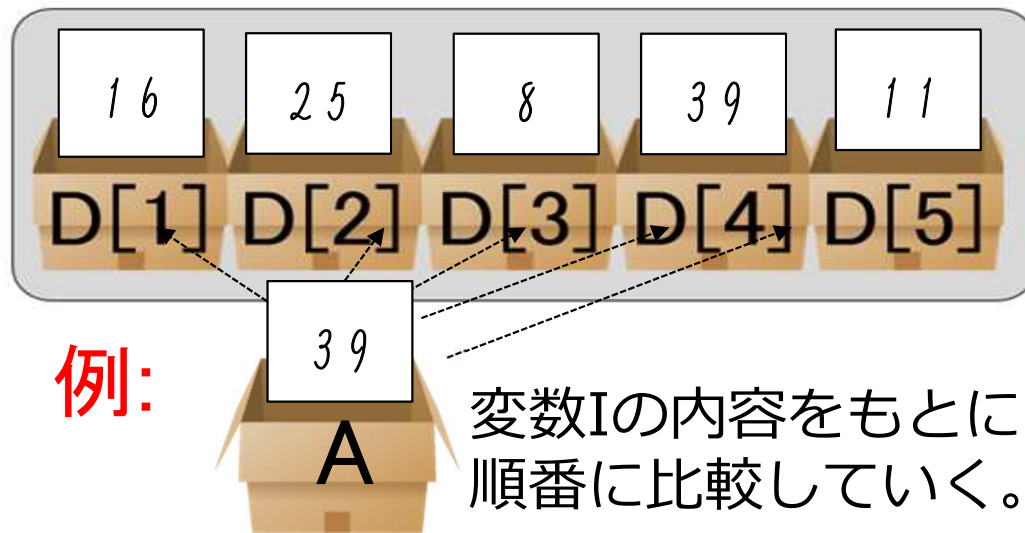
プログラムで、探す数がリストDの何番目に入っているか変数Xに入れます。

## 課題5補足:配列の中から数を探す

ヒント2

プログラムの中で数の探し方

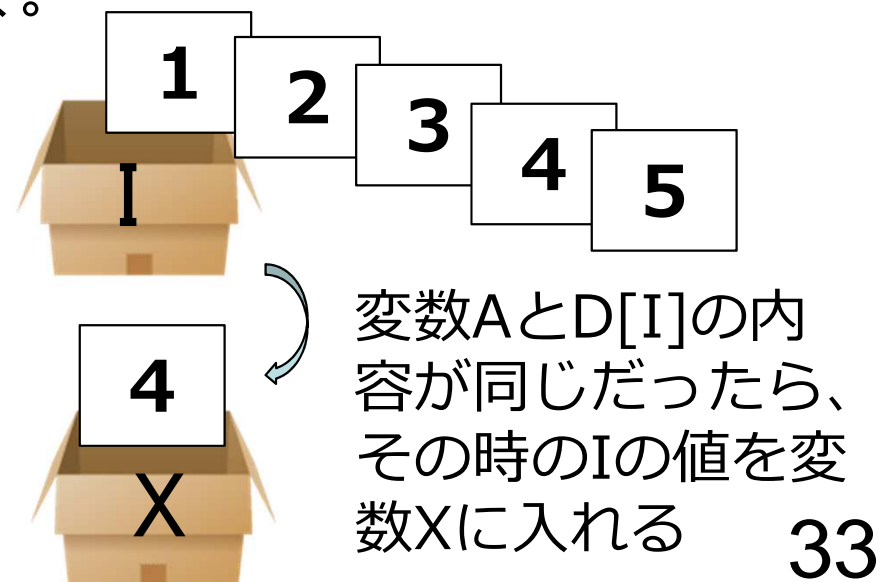
変数Iを1～5まで変えていき、D[I]とAの内容を比較します。



例:

変数Iの内容をもとに  
順番に比較していく。

もし等しければ、I(何番  
目かの番号)の値をXに  
入れます。



## 課題5補足: リストの中から数を探す(検索)

ヒント3

D[]に5個の数をいれとく

A = (入力)

X = 0

I = 1, 2, 3...  
I = 5まで

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

Xを表示

部品  
02

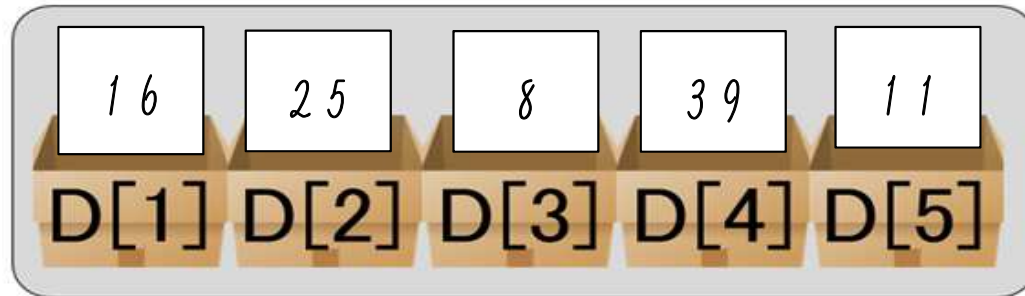
この部分を考えて、プログラムを  
完成されてください。

部品  
03

部品  
07

## 課題6: リストの中の一番小さい数を見つける

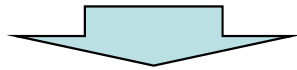
こんな動作をするプログラムを作ります



**ヒント:**

打ち込み5: リストを使った5つの数の合計」のプログラムを流用して(コピーして保存)して作る。

初めにリストに数を入れておきます



8 です。

プログラムで、一番小さな数を探します。



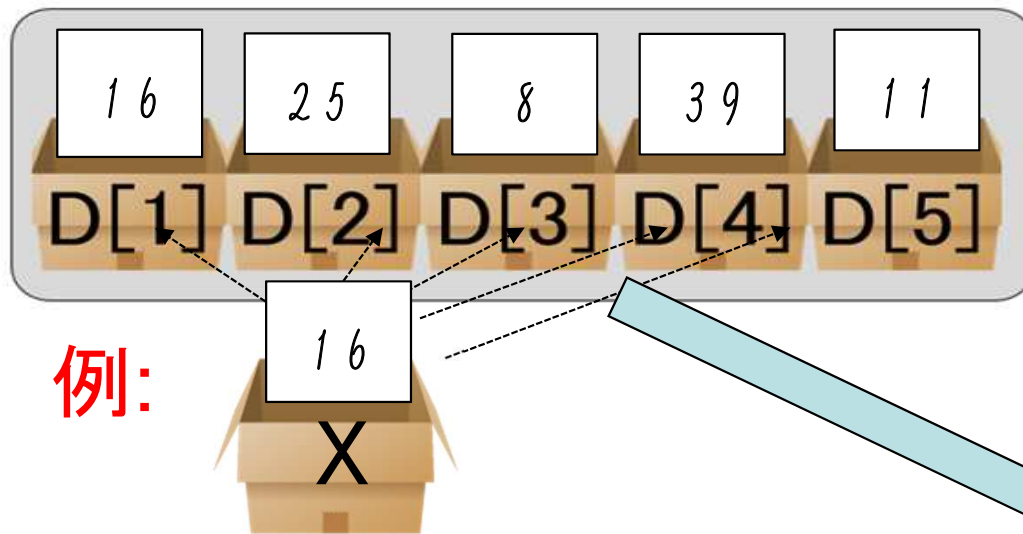
変数としては変数Xに、一番小さな数が入るようにします。

次のヒント1~2のスライドも見て

## 課題6: リストの中の一番小さい数を見つける

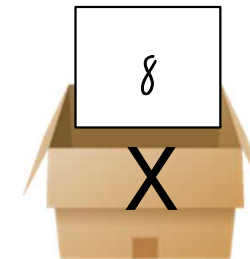
ヒント1

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。



例:

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



## 課題6補足: リストの中の一番小さい数を見つける

ヒント2

D[]に5個の数をいれとく

$X = D[1]$

$I = 1, 2, 3 \dots$   
 $I = 5$ まで

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

Xを表示

ポイント:

変数Xに初め0が入っていると他の数より小さいので初めにD[1]の値を入れています。

この部分を考えて、プログラムを完成されてください。

部品  
03

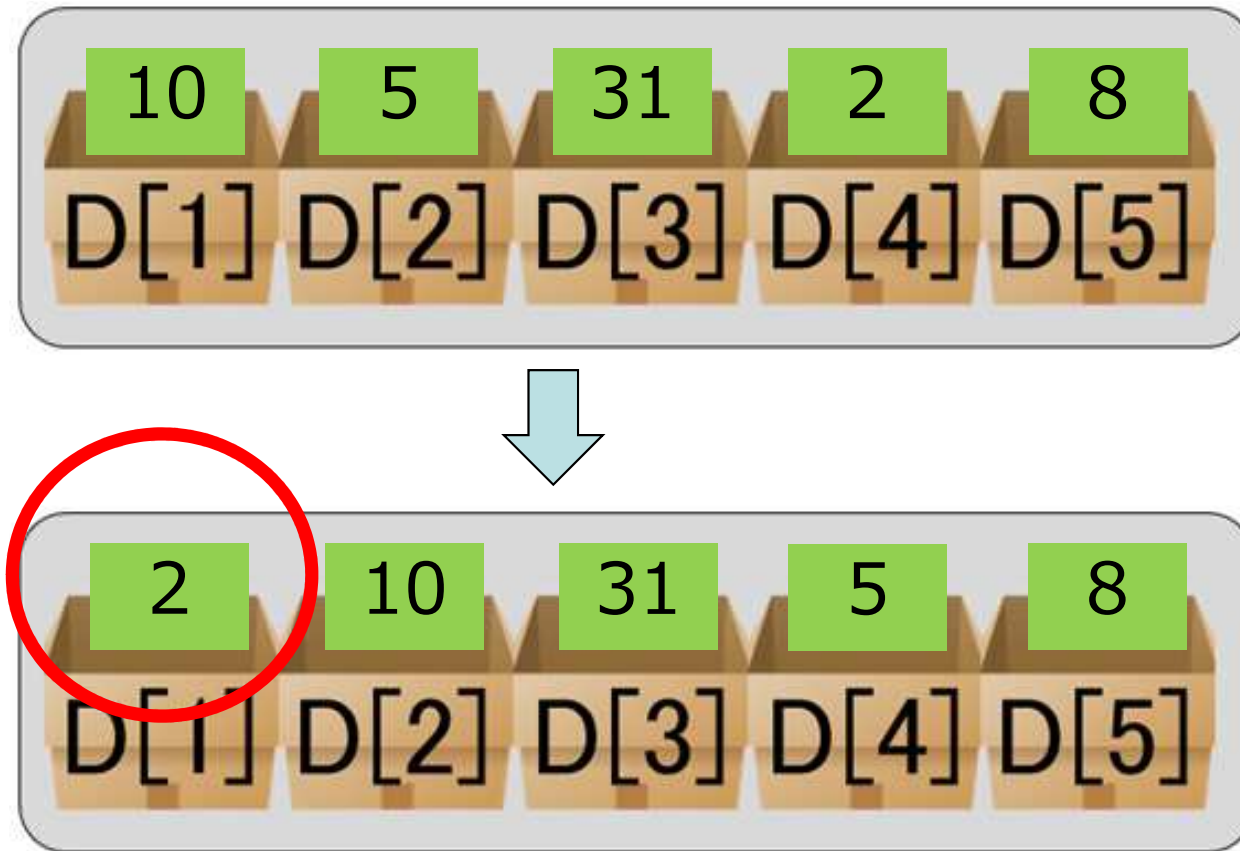
部品  
07

部品  
09

## 課題7: 一番小さい数を配列の先頭に入れ替える

こんな動作をするプログラムを作ります

開発



ヒント:

課題6: リストの中の一  
番小さい数を見つける  
のプログラムを流用して  
(コピーして保存)して作る。

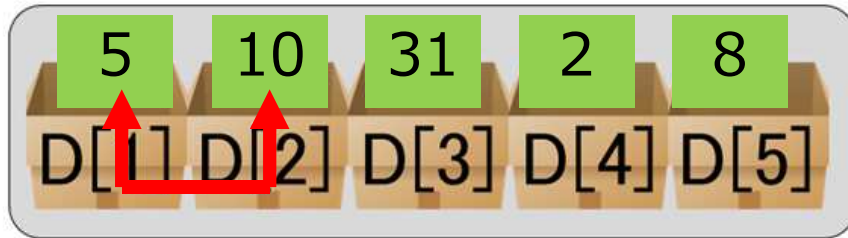
一番小さい数がD[1]に入ります。但し、リストには元のリストにあったすべての数が残っています。

次のヒント1~3  
のスライドも  
見て

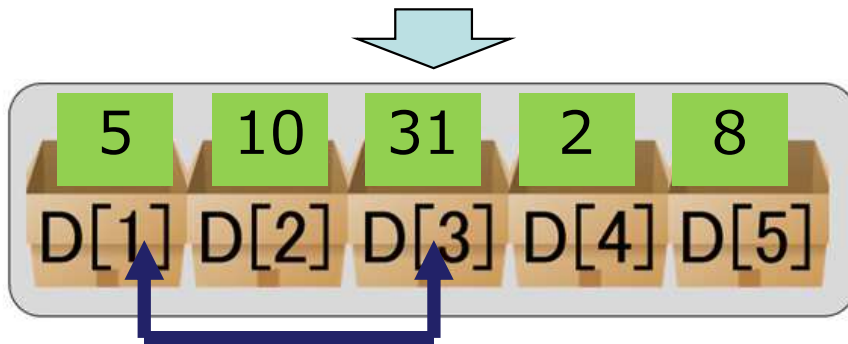
# 課題7補足：一番小さい数を配列の先頭に入れ替える

プログラムの考え方(紙に数値を書いてやってみよう)

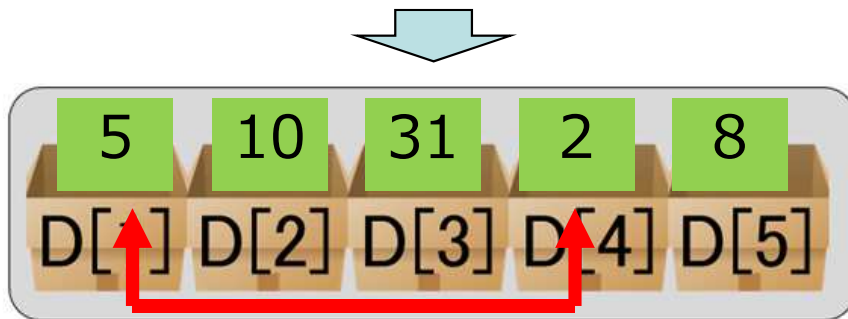
ヒント1



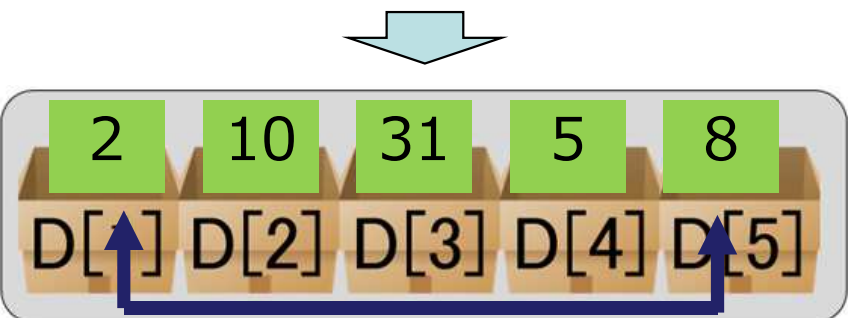
1番目と2番目を比較して5の方が小さいので入れ替えない



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



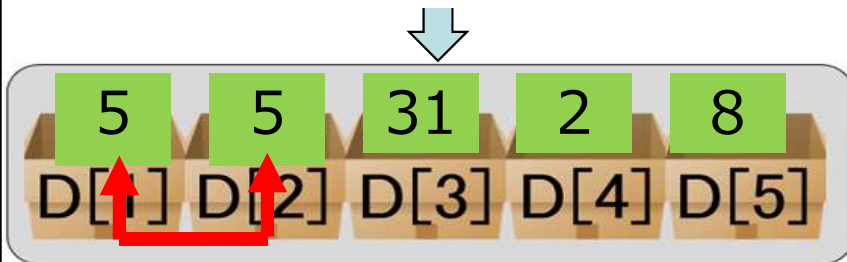
1番目と5番目を比較して2の方が小さいので入れ替えない

# 課題7補足：一番小さい数を配列の先頭に入れ替える 数の入れ換え

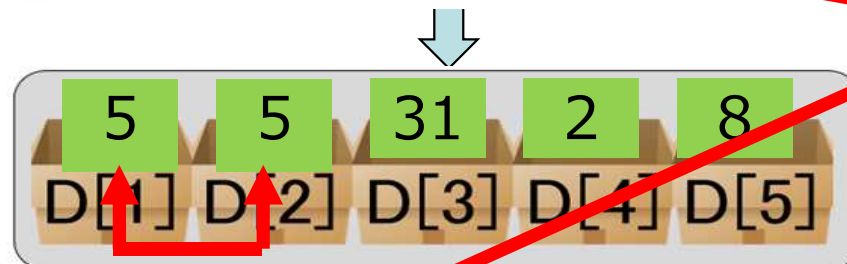
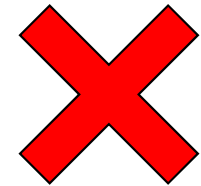
ヒント2



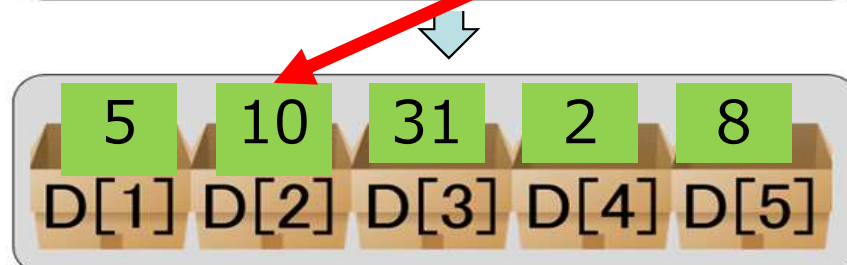
1番目と2番目の入れ替え



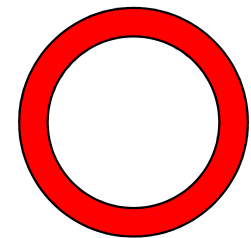
いきなり入れ替えると  
10が残らない



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



部品  
08



## 課題7補足：一番小さい数を配列の先頭に入れ替える

ヒント3

D[]に5個の数をいれとく

I = 1, 2, 3...  
I = 5まで

変数D[1]とD[I]の内容を比較して、D[I]が小さければ、その時の値をD[1]に入れる

この部分を考えて、プログラムを完成させてください。

部品  
03

部品  
08

部品  
08

部品  
09

## 打ち込み6:二重繰り返しで九九に挑戦

理解・打ち込み

二重繰り返しを使って、九九を言うプログラムを作ってみましょう



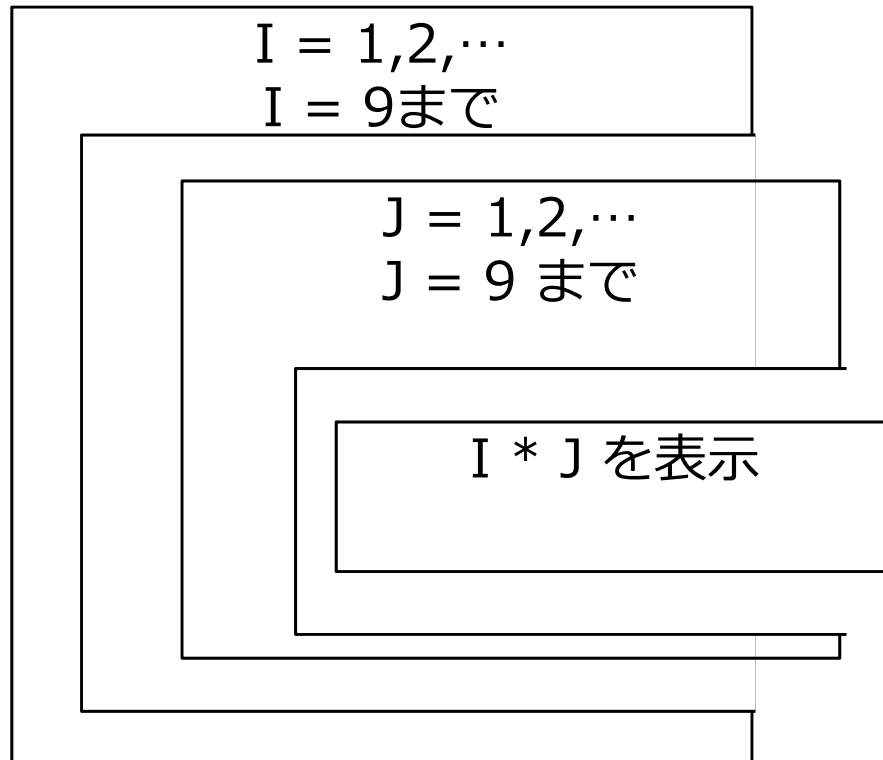
1, 2, 3, 4, 5, 6, 7, 8, 9  
2, 4, 6, 8, 10, 12, 14, 16, 18  
3, 6, 9, 12, 15, 18, 21, 24, 27  
.....  
9, 18, 27, 36, 45, 54, 63, 72, 81

1 x 1, 1 x 2, 1 x 3, 1 x 4, 1 x 5, 1 x 6, 1 x 7, 1 x 8, 1 x 9  
2 x 1, 2 x 2, 2 x 3, 2 x 4, 2 x 5, 2 x 6, 2 x 7, 2 x 8, 2 x 9  
3 x 1, 3 x 2, 3 x 3, 3 x 4, 3 x 5, 3 x 6, 3 x 7, 3 x 8, 3 x 9  
.....  
9 x 1, 9 x 2, 9 x 3, 9 x 4, 9 x 5, 9 x 6, 9 x 7, 9 x 8, 9 x 9

次のヒントの  
スライドも見て

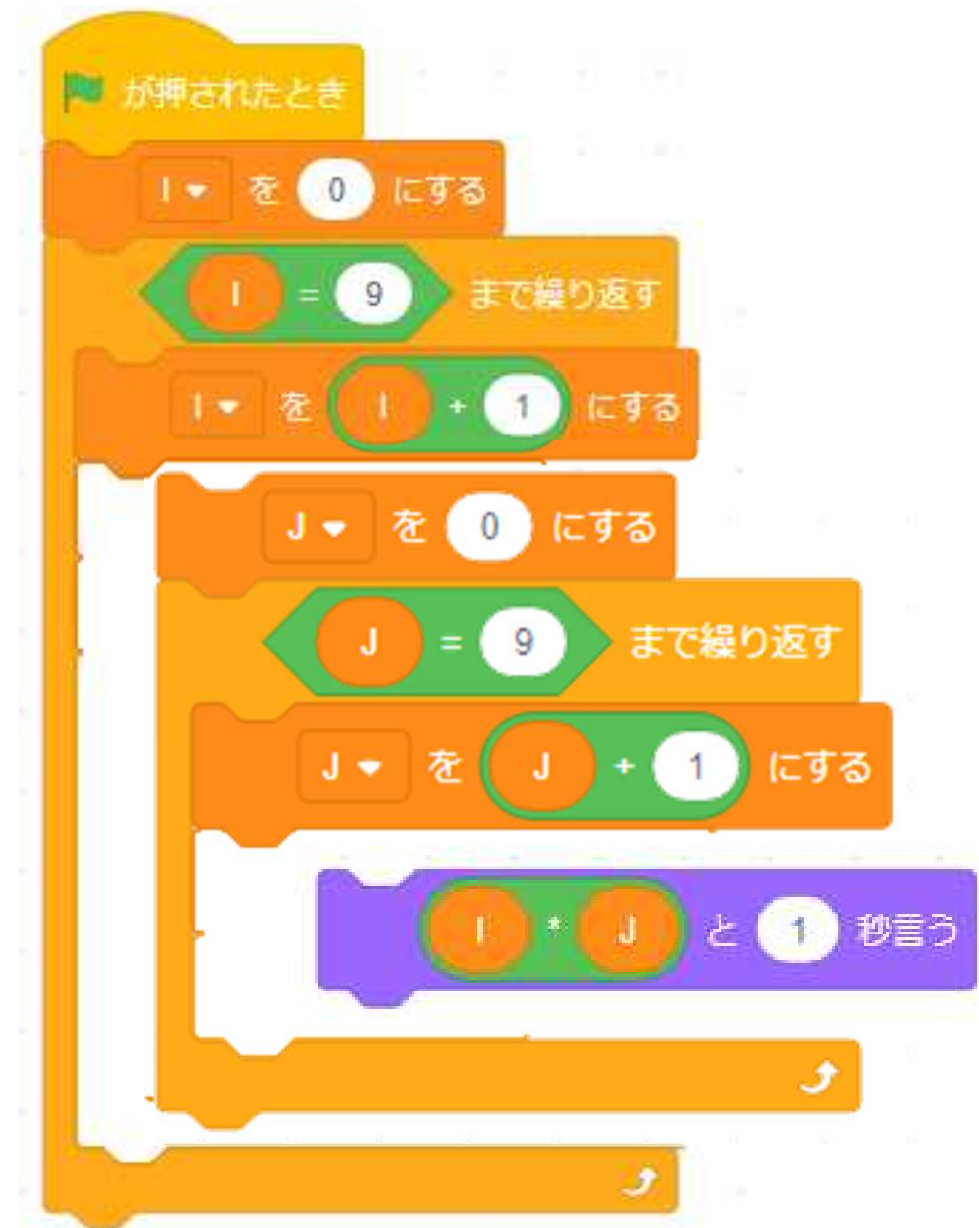
# 打ち込み:二重繰り返しに挑戦

ヒント



部品  
10

そのまま打ち込む  
プログラム  
(間はあけない)



# 打ち込み7:複雑な二重繰り返しに挑戦

理解・打ち込み

二重繰り返しですが、内側の繰り返しの始まりの数を変えます。  
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  
．．．．  
最期にD[5]までの数を言うプログラムを作ります。

例えば、D[1]～[5]までに、  
12, 9, 65, 87, 44を入れておいた時



12, 9, 65, 87, 44  
9, 65, 87, 44  
65, 87, 44  
87, 44  
44



部品  
11

次のスライドにプログラムと図式。  
次の次のスライドに解説があります。

次のヒント1～2の  
スライドも見て

# 打ち込み:複雑な二重繰り返しに挑戦

ヒント1

D[]に5個の数をいれとく

$I = 1, 2, \dots$

$I = 5$ まで

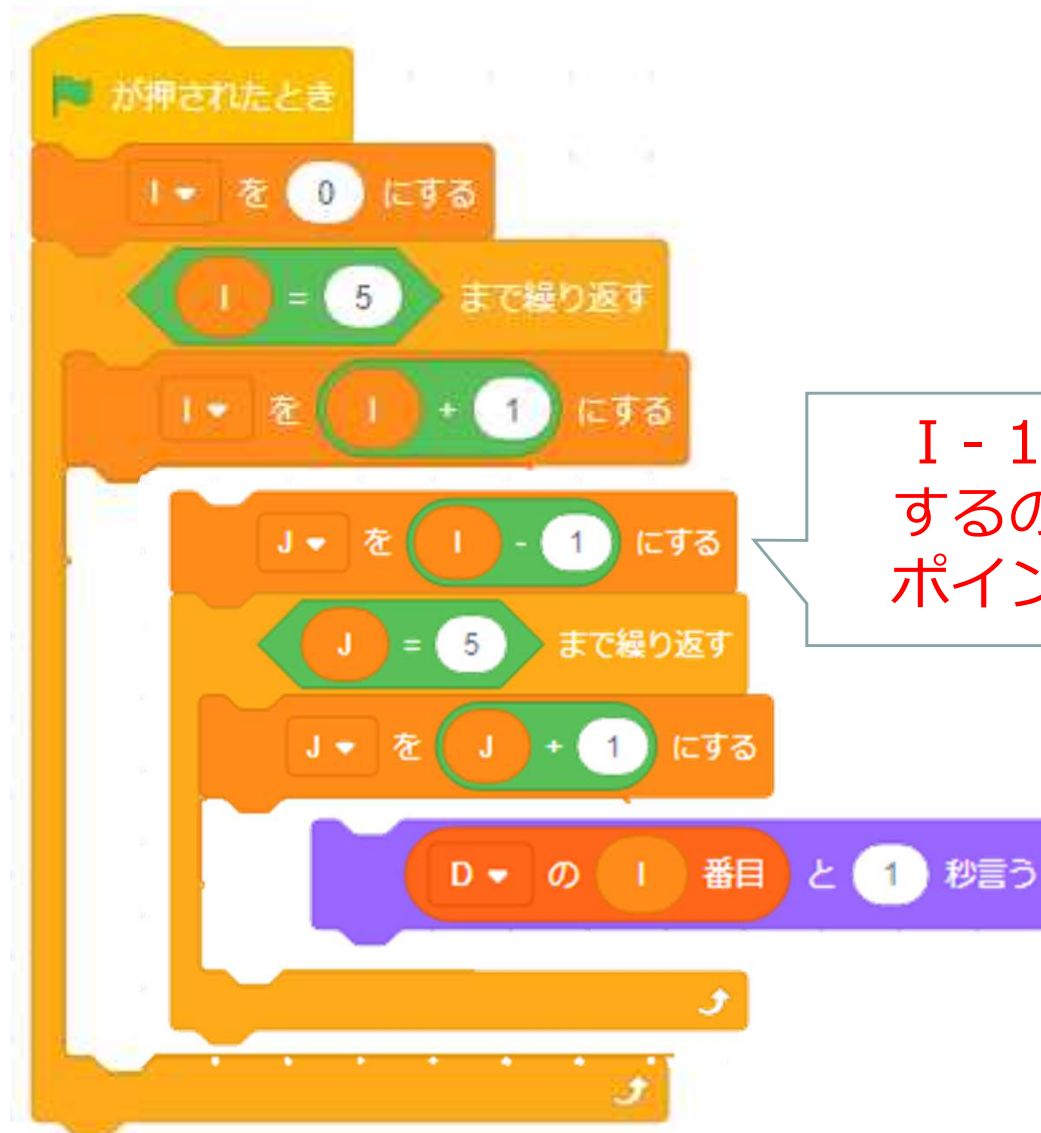
$J = I, I + 1, \dots$

$J = 5$ まで

D[I]を表示

部品  
09

部品  
11



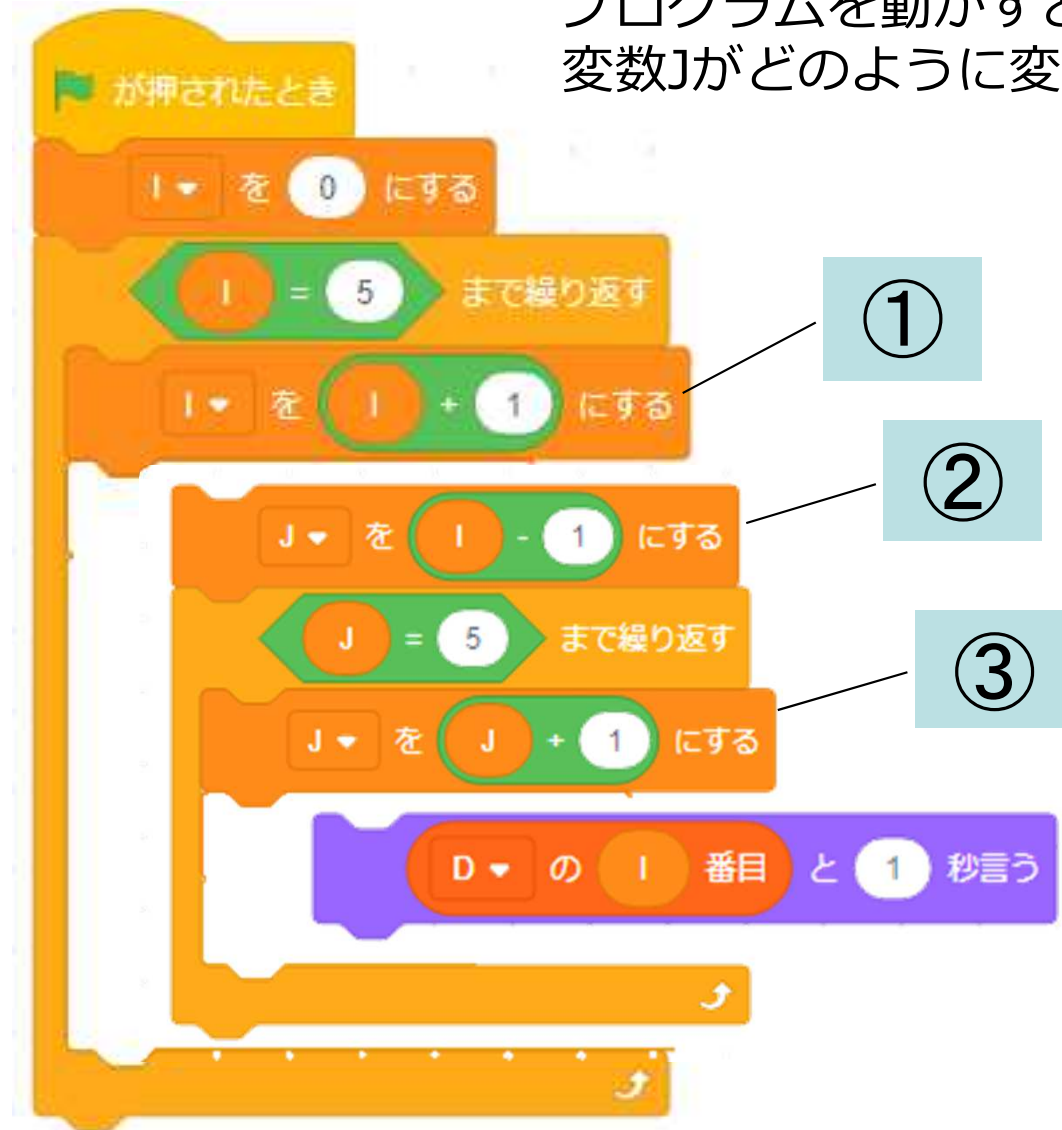
I - 1に  
するのが  
ポイント

そのまま打ち込む  
プログラム(間はあけない)

# 複雑な二重繰り返しに挑戦

ヒント2

プログラムを動かすと①②③のどこを通過して、変数Iと変数Jがどのように変わるか考えてみましょう。

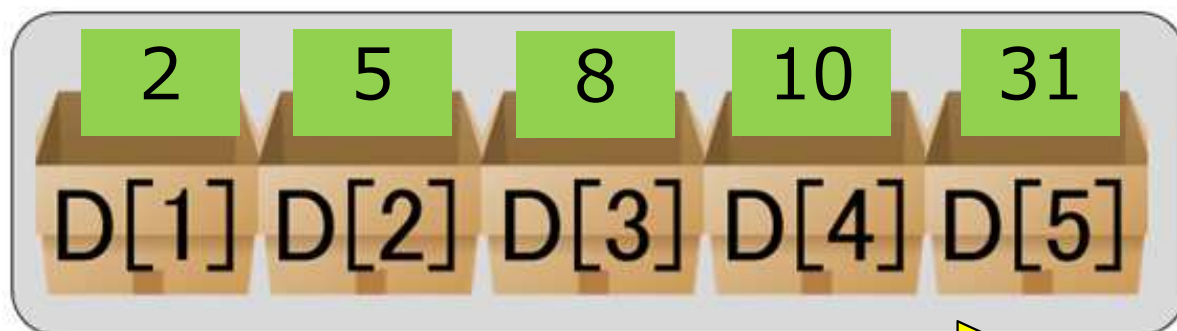
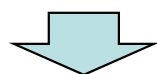
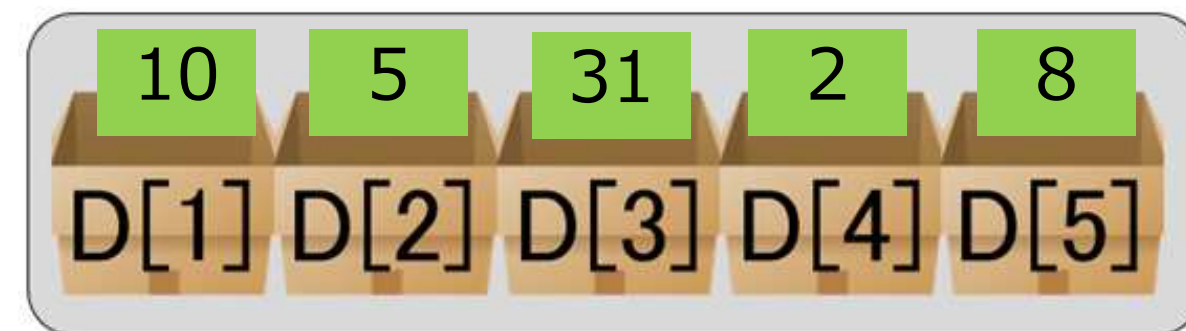


通る場所	I	J
①	1	?
②	1	0
③	1	1
③	1	2
③	1	3
③	1	4
③	1	5
①	2	5
②	2	1
③	2	2
③	2	3
③	2	4
続いていきます		

## 課題8: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。

次のヒント1~2のスライドも見て

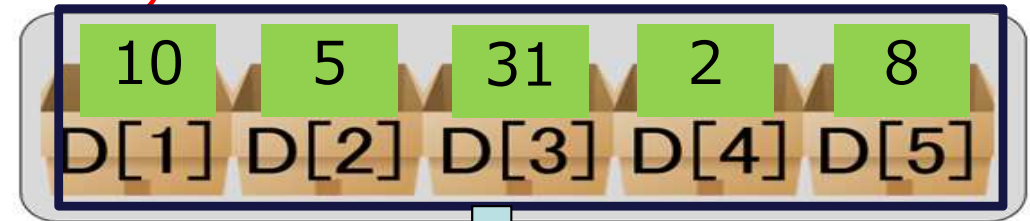
# 最期のチャレンジ:数の並び替え

ヒント1

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)

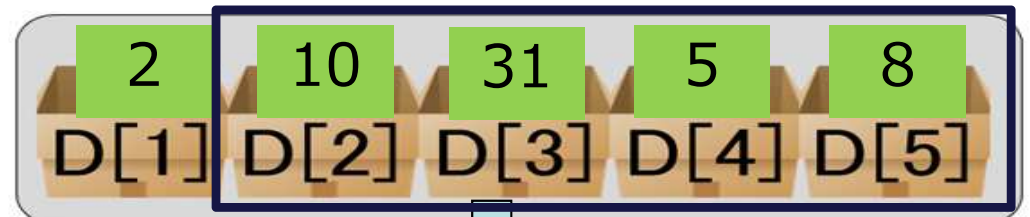
1回目

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 48

# 最期のチャレンジ:数の並び替え

ヒント2

D[]に5個の数をいれとく

$I = 1, 2, \dots$   
 $I = 5$ まで

$J = I, I + 1, \dots$   
 $J = 5$ まで

変数D[I]に一番小さい数が入るようにする処理

ヒント:

打ち込み7:複雑な二重繰り返しに挑戦のプログラムを流用して(コピーして保存)して作る。

この部分を考えて、プログラムを完成させてください。  
課題7のこのあたりのプログラムを少し変えるだけでできます。

部品  
03

部品  
08

部品  
08

部品  
09

部品  
11

# 発展課題1: FizzBuss

開発

Fizz Buzzという遊びのプログラムを作ってみましょう。  
例えば、1から30までの数を順番に言いますが。

3の倍数の時は"Fizz"と言います。

5の場合の時は"Buzz"と言います。

3と5の倍数の時は、"Fizz Buzz"と言います。

それ以外は数字と言います。

## 考え方

- ・まず1から30までの数をつくる構造を考えます。
- ・その中で、3の倍数かどうか判断します。3の倍数で無ければ、5の場合数か判断します。5の倍数の場合は、"Buss"といいいます。3の場合数でも5の倍数でもなければ数字をいいいます。
- ・3の倍数だった場合は、5の倍数か判断して、5の倍数でもあったら"Fizz Buzz"と言います。5の倍数でなかったら3の倍数なので"Fizz"と言います。

## 発展課題2: バブルソート

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くつもあります。

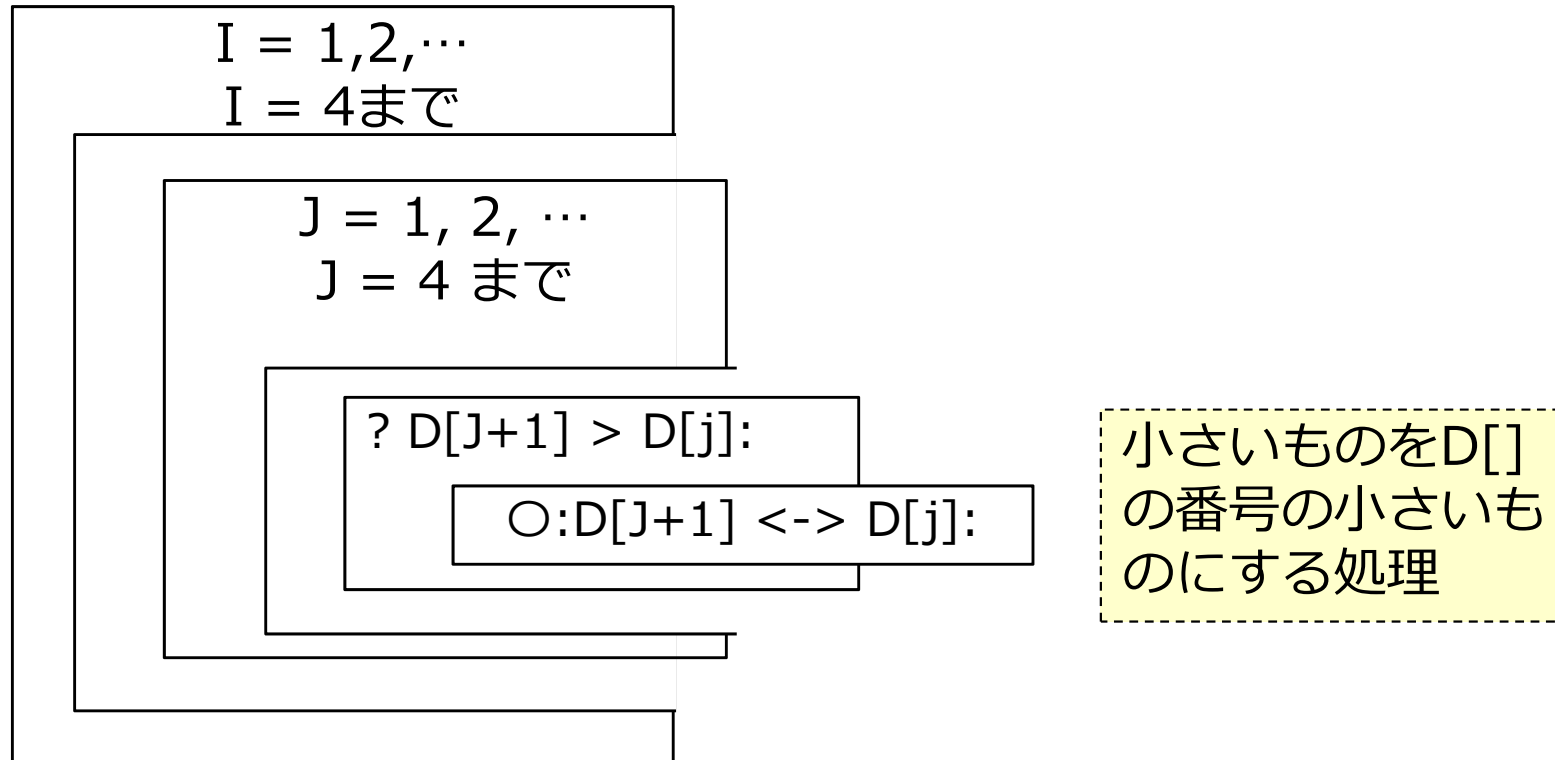
次の並び替え方法の考え方やプログラムをWebで調べて作成してください。

バブルソート

次のヒントの  
スライドも見て

# バブルソートの図式

D[]に5個の数をいれとく



部品  
08

部品  
09

部品  
10

## 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/バブルソートで処理速度がどのように違うか測定してみましょう。

タイマーをリセット

ここに並び替えをするプログラムを入れます

A ▼ を タイマー にする

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。

次のヒントの  
スライドも見て

3



# 配列に数をセットする方法

ランダムな数を設定する。たくさん  
の数を設定する時に良い

ヒント

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。



部品  
09

学習終了時のアンケート

学習終了時のアンケート

<https://forms.gle/>