

2025年度共通テスト「情報」  
第3問  
スクラッチを使って解説します！



ユーバー プログラミング スクール



ユーバー プログラミング スクール

# 目的

大学共通テスト「情報」問題の中から選び、

高校生はもちろん、中学や小学高学年の方にも

『楽しく』『簡単に』理解できて

問題を解けるようにする動画です！

ゆる～く楽しみながら情報を少し学びましょう！



## 第3問

ある生徒が所属する  
「工芸部の集中制作合宿」での  
**担当決めを自動化する問題**です。

お手元に問題がある方はじっくりと読んでみてく  
ださいね。

問1から順番に見ていきましょう！



# 問1

図や表を適切に読み、前提を理解する



# 問1

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

作るのは**9種類の工芸品**です。

それぞれ**作るのに何日かかるのか**、表1を見るとわかります。

部員の**誰が作ってもかかる日数は同じ**です。

部員は1つの工芸品を作っている間に別のものは作りません。**(その期間は1つだけ)**

作る工芸品は、**1から順番に部員に割り当て**ます。



# 問1

出題

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

問題)工芸品4は、誰が何日目から1日間で作成するでしょうか？



# 問1

答え合わせ

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

解答)工芸品4は、「部員2」が2日目から1日間で作成します！



# 問1

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

図1には工芸品4までが割り当てられています。部員1は5日目に割り当てがなくなり「空き(あき)」になります。そこで、次のルールで続きを割り当てることにするそうです！

最も早く空きになる部員(複数の時は部員番号が一番小さい人)が「空き」になった日から次の工芸品を作る！



# 問1

お知らせするメールの文章(図2:各工芸品の担当と期間を一覧にしたメールの文面)

工芸品 1 ... 部員 1 : 1日目～4日目  
 工芸品 2 ... 部員 2 : 1日目～1日目  
 工芸品 3 ... 部員 3 : 1日目～3日目  
 工芸品 4 ... 部員 2 : 2日目～2日目  
 工芸品 5 ... **部員 ? : ? 日目～ ? 日目**

工芸品 9 ... 部員 1 : 7日目～9日目

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

図2のように担当と制作期間をメールでお知らせするそうです。

出題では、工芸品5は**誰が何日目から何日目**まで作成するのかメールの文面を埋める数字を問われています。次から考えていきましょう。



# 問1

出題

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

問題)工芸品5は、誰が何日目から何日目まで作成するでしょうか？



# 問1

答え合わせ

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4	(工芸品)5								
部員3		(工芸品)3									

解答)工芸品5は、「部員2」が3日目から5日目まで作成します！

一番早く「空き」になるのは部員2だもんね！



## 問2

誰が次に「空き」になるのか  
プログラムを完成する



# 問2

お知らせするメールの文章(図2:各工芸品の担当と期間を一覧にしたメールの文面)

工芸品 1 ... 部員 1 : 1日目～4日目  
 工芸品 2 ... 部員 2 : 1日目～1日目  
 工芸品 3 ... 部員 3 : 1日目～3日目  
 工芸品 4 ... 部員 2 : 2日目～2日目  
 工芸品 5 ... **部員 2 : 3日目～5日目**

工芸品 9 ... 部員 1 : 7日目～9日目

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表ー工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

図2のように担当と制作期間をメールでお知らせするけれど、、、

めんどいですね。。。そこで自動的に担当(次の空き)がわかるように「プログラミング」していきます！



おさえおこう！

# 共通テストを解くために

## 変数の使い方

buinsu を 3 にする

tantou を buin にする

buinsu = 3

tantou = buin

## 配列(スクラッチの場合リスト)の使い方

Akibi のすべてを削除する

5 を Akibi に追加する

3 を Akibi に追加する

4 を Akibi に追加する

Akibi の 1 番目

Akibi の tantou 番目

Akibi = [5,3,4]

Akibi [1]

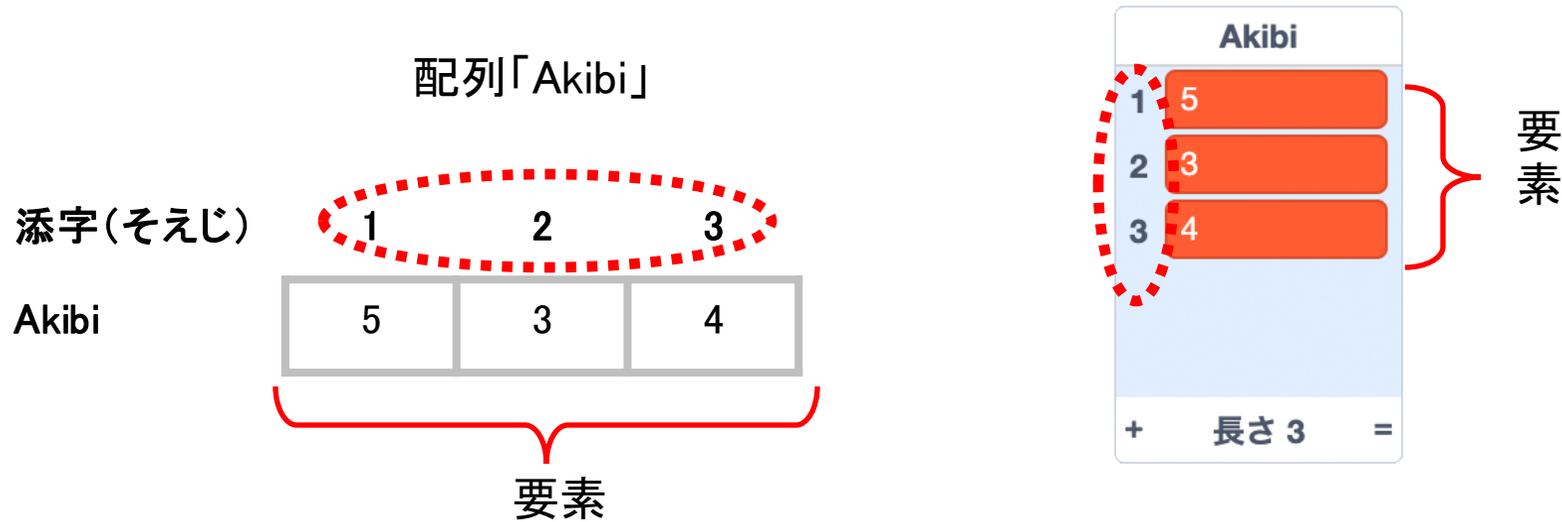
Akibi [tantou]

(そえじ)  
添字の指定



# 添字(そえじ)、要素ってなんだ???

共通テストの問題は、配列の添字(そえじ)や要素という難しい言葉がでてきたね！



「添字」は、スクラッチのリストだと、1番目、2番目、3番目…の数字と同じようなものだよ。配列(リスト)には**たくさんのデータ**を入れることができるから、何番目のデータなのか、わかるようにするんだよね！**データ**のことを「要素」というよ。

## では、問題に戻ろう！



# 問2

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

まずは誰が一番はやく「空き」になるかわかるように「配列 Akibi(スクラッチでは、リストだよ)」を準備する。

Akibiの添字(1番目、2番目、3番目)を「部員の番号」にするよ。

部員1はAkibiの1番目に、部員2はAkibiの2番目に、空き日を入れる。

添字(そえじ)

1 2 3

Akibi

5	3	
---	---	--

(図3:図1に対応する配列「Akibi」)

Akibi

1	5
2	3
3	

+ 長さ 3 =



# 問2



誰が何をいつ作るのか(図1:割り当て表—工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

問題)リスト「Akibi」の3番目には何が入るかな？

添字(そえじ)

1

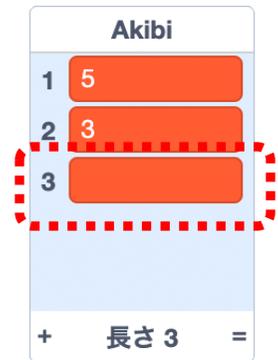
2

3

Akibi

5

3



(図3:図1に対応する配列「Akibi」)



# 問2



誰が何をいつ作るのか(図1:割り当て表—工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

解答)リスト「Akibi」の3番目には  
4が入るよ

→部員3の空きは4日目だよね!

添字(そえじ)	1	2	3
Akibi	5	3	4

(図3:図1に対応する配列「Akibi」)

Akibi	
1	5
2	3
3	4
+ 長さ 3 =	



# 問2



誰が何をいつ作るのか(図1:割り当て表—工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

問題) 配列「Akibi」から一番早く「空き」になる部員を見つける。

「最小の値」は「部員(?)」なので一番早く「空き」になるのは?

添字(そえじ)

1

2

3

Akibi

5

3

4

Akibi	
1	5
2	3
3	4
+ 長さ 3 =	

(図3: 図1に対応する配列「Akibi」)



# 問2



答え合わせ

誰が何をいつ作るのか(図1:割り当て表一工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

解答)配列「Akibi」から一番早く「空き」になるのは

「部員2」

なぜなら配列「Akibi」の値で最小は2番目の3だからだね!

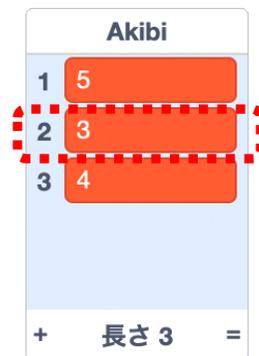
添字(そえじ)

1 2 3

Akibi

5 3 4

(図3:図1に対応する配列「Akibi」)



## 問2

この考え方(配列の要素で最小)で次の担当者を表示するプログラムを作ろう。

- ①配列「Akibi」に5、3、4を入れる(図1に基づいた空き日)
- ②変数「buinsu」に3を入れる  
変数「tantou」に1を入れる  
変数「buin」に2を入れる → 最初は2から始めるため
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで③を繰り返す



# 共通テストを解くために

使用する配列と変数を整理しておくのが大切！

配列(リスト)	
Akibi	部員の次の空き日を入れる。添字は部員番号

変数	
buinsu	部員の総数(全部で何人か)
buin	部員番号を指定するために使う。例) Akibiのbuin番号番目
tantou	担当部員の番号を指定するために使う。Akibiのtantou番号番目



# プログラムで考えよう

スクラッチでやってみよう！

- ①リスト「Akibi」に部員1～3の次の空き日を入れよう。
- ②変数「buinsu」「tantou」「buin」に最初の値を入れよう。



## 問2 続きを考えよう！

- ①配列「Akibi」に5、3、4を入れる(図1に基づいた空き日)
- ②変数「buinsu」に3を、変数「tantou」に1を入れる  
変数「buin」に2を入れる

Akibi	
1	5
2	3
3	4
+ 長さ 3 =	

- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、最後の部員まで調べる

②で入れた最初の値	tantou 1	buin 2
「Akibi」の「tantou」番目(最初は1番目)と「buin」番目(最初は2番目)を比べて早い「空き」(小さい方の番号)を「tantou」にしておく	tantou 2	
「buin」を1増やす		buin 3
「buin」が「buinsu」になるまで繰り返す	tantou 2	
「Akibi」の「tantou」番目(ここでは2)と「buin」番目(ここでは3番目)を比べて早い「空き」(小さい方の番号)を「tantou」にしておく		



# プログラムで考えよう

- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する

「Akibi」の「tantou」番目と「buin」番目を比べて  
早い「空き」(小さい方の番号)を「tantou」にしておく  
「buin」を1ずつ増やす

変数「buin」を2から1ずつ増やして、最後の部員まで調べる

変数「tantou」を「buin」にする条件から考えてみよう

Akibi ▾ の buin 番目 と Akibi ▾ の tantou 番目 を比べてどうだったら

tantou ▾ を buin にする にするのかな？



スクラッチでやってみよう！

# プログラムで考えよう

- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する

「Akibi」の「tantou」番目と「buin」番目を比べて  
早い「空き」(小さい方の番号)を「tantou」にしておく  
「buin」を1ずつ増やす

Akibi	
1	5
2	3
3	4
+ 長さ 3	=

変数「buin」を2から1ずつ増やして、最後の部員まで調べる

変数「tantou」を「buin」に代入するのは、配列Akibiのbuin番目が配列Akibiのtantou番目より小さい(空き日が早い)時だね！

条件はこんなふうになるよね！



スクラッチでやってみよう！

# プログラムで考えよう

- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する

もし  $\text{Akibi の buin 番目} < \text{Akibi の tantou 番目}$  なら  
tantou を buin にする

Akibi	
1	5
2	3
3	4
+ 長さ 3	=



# プログラムで考えよう

- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、最後の部員まで調べる

「Akibi」の「tantou」番目と「buin」番目を比べて  
早い「空き」(小さい方の番号)を「tantou」にしておく

「buin」を1ずつ増やす

「buin」が「buinsu」(ここでは3)より大きくなるまで繰り返す

変数「buin」を1増やすには、スクラッチは次のブロックを使うよ！



# プログラムで考えよう

スクラッチでやってみよう！

④変数「buin」を2から1ずつ増やして、最後の部員まで調べる(③を繰り返す)



# プログラムで考えよう

- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、最後の部員まで調べる(③を繰り返す)

「Akibi」の「tantou」番目と「buin」番目を比べて  
早い「空き」(小さい方の番号)を「tantou」にしておく

「buin」を1ずつ増やす

- ④変数「buin」を2から1ずつ増やして、最後の部員まで調べる

変数「buin」が「buinsu」より大きくなるまで 表の中のプログラムを繰り返すよ！



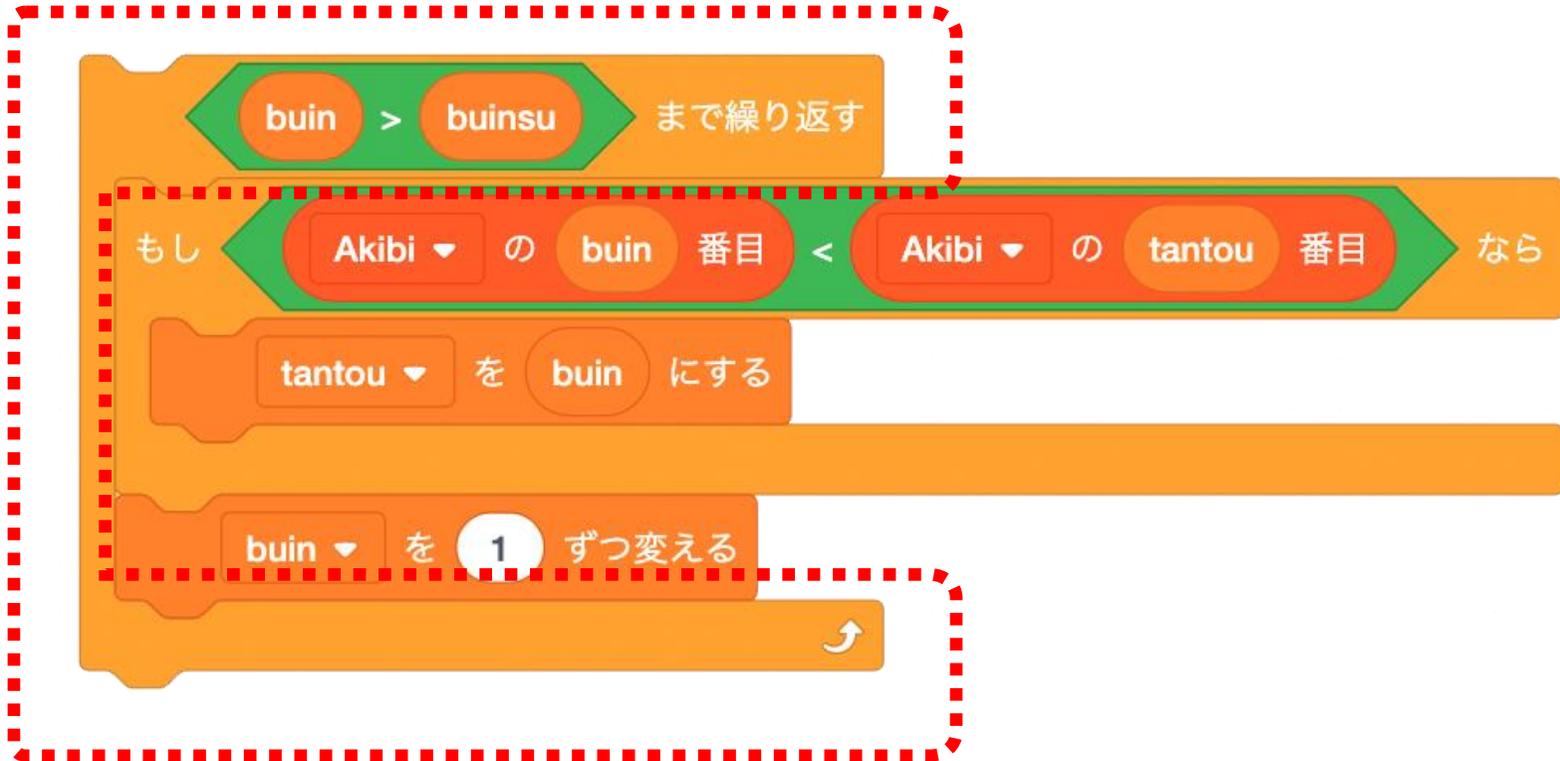
条件に合わない間にはさんだ  
ブロックを繰り返すよ！



# プログラムで考えよう

スクラッチでやってみよう！

④変数「buin」を2から1ずつ増やして、最後の部員まで調べる



# プログラムで考えよう

①～④を繋げて  
セリフで言う。

スクラッチでやってみよう！

定義 問2

Akibi ▾ のすべてを削除する

5 を Akibi ▾ に追加する

3 を Akibi ▾ に追加する

4 を Akibi ▾ に追加する

buinsu ▾ を 3 にする

tantou ▾ を 1 にする

buin ▾ を 2 にする

buin > buinsu まで繰り返す

もし Akibi ▾ の buin 番目 < Akibi ▾ の tantou 番目 なら

tantou ▾ を buin にする

buin ▾ を 1 ずつ変える

次の工芸品の担当は部員 と tantou と です。 という



# 問2

出題

誰が何をいつ作るのか (図4: 次に割り当てる工芸品の担当部員を表示するプログラム)

Akibi = [5, 3, 4]

buinsu = 3

tantou = 1

buin を 2 から buinsu まで1ずつ増やしながら繰り返す:

もし ? ならば:

tantou = buin

表示する ("次の工芸品の担当は部員", tantou, "です。")

問題) ? にあてはまる条件を選ぶ。

buin < tantou

buin < tantou

buin > tantou

buin > tantou

Akibi の buin 番目 < Akibi の tantou 番目

Akibi[buin] < Akibi[tantou]

Akibi の buin 番目 > Akibi の tantou 番目

Akibi[buin] > Akibi[tantou]



# 問2

答え合わせ

誰が何をいつ作るのか (図4: 次に割り当てる工芸品の担当部員を表示するプログラム)

```
Akibi = [5, 3, 4]
```

```
buinsu = 3
```

```
tantou = 1
```

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし ? ならば:

```
tantou = buin
```

表示する ("次の工芸品の担当は部員", tantou, "です。")

変数「tantou」を「buin」に代入するのは、配列Akibiのbuin番目が配列Akibiのtantou番目より小さい(空き日が早い)時だね！



$Akibi[buin] < Akibi[tantou]$



# 問2

部員数が5人になった場合を考えよう。配列「Akibi」の要素は[5, 6, 4, 4, 4]とする。

- ①配列「Akibi」に5, 6, 4, 4, 4を入れる(出題に基づいた空き日)
- ②変数「buinsu」に5を入れる  
変数「tantou」に1を入れる  
変数「buin」に2を入れる → 最初は2から始めるため
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで③を繰り返す



出題

## 問題

の部分で変数「tantou」へ部員番号(「buin」)を代入する回数は何回でしょうか？



## 問2 考えてみよう！

- ①配列「Akibi」に5, 6, 4, 4, 4を入れる(出題に基づいた空き日)
- ②変数「buinsu」に5を、変数「tantou」に1を入れる  
変数「buin」に2を入れる
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで(ここでは5) ③を繰り返す

Akibi	
1	5
2	6
3	4
4	4
5	4
+	長さ 5 =

	tantou	buin
②で入れた最初の値	1	2

「buin」が「buinsu」より小さい間は「buin」を1ずつ増やして以下を繰り返す

	回数	実行後 tantou	buin	代入
③「Akibi」の「tantou」番目と「buin」番目を比べて もし「Akibi」の「buin」番目の「空き日」が早い(小さい)なら「buin」を「tantou」に代入する	1回目	1	2	
	2回目	3	3	○
	3回目	3	4	
	4回目	3	5	



# 問2

部員数が5人になった場合を考えよう。配列「Akibi」の要素は[5, 6, 4, 4, 4]とする。

- ①配列「Akibi」に5, 6, 4, 4, 4を入れる(出題に基づいた空き日)
- ②変数「buinsu」に5を入れる  
変数「tantou」に1を入れる  
変数「buin」に2を入れる → 最初は2から始めるため
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで③を繰り返す

答え合わせ

## 問題



の部分で変数「tantou」へ部員番号(「buin」)を代入したのは

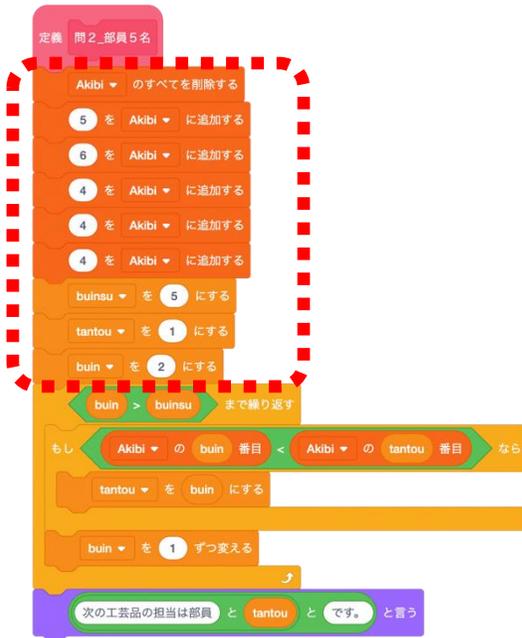
1回です！



# プログラムでたしかめよう

- ①配列「Akibi」に5, 6, 4, 4, 4を入れる(出題に基づいた空き日)
- ②変数「buinsu」に5を、変数「tantou」に1を入れる  
変数「buin」に2を入れる
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで③を繰り返す

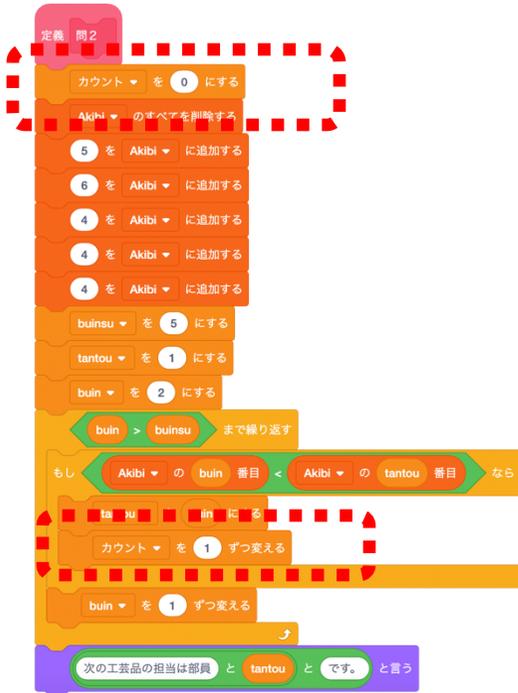
Akibi	
1	5
2	6
3	4
4	4
5	4
+ 長さ 5 =	



# プログラムでたしかめよう

- ①配列「Akibi」に5, 6, 4, 4, 4を入れる(出題に基づいた空き日)
- ②変数「buinsu」に5を、変数「tantou」に1を入れる  
変数「buin」に2を入れる
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで③を繰り返す

Akibi	
1	5
2	6
3	4
4	4
5	4
+ 長さ 5 =	



出題にはないけれど  
何回代入したか  
数えてみます！



# プログラムでたしかめよう

- ①配列「Akibi」に5, 6, 4, 4, 4を入れる(出題に基づいた空き日)
- ②変数「buinsu」に5を、変数「tantou」に1を入れる  
変数「buin」に2を入れる
- ③[Akibi]の要素(データ)と変数を使って早く空きになる担当部員を順番に調べる  
→変数「tantou」に早く空きになる部員番号(「buin」)を代入する
- ④変数「buin」を2から1ずつ増やして、「buinsu」まで③を繰り返す

Akibi	
1	5
2	6
3	4
4	4
5	4
+ 長さ 5 =	

ユーパープログラミングスクール

次の工芸品の担当は部員3です。

buin 6    tantou 3

カウント 1

出題にはないけれど  
何回代入したか  
数えてみます！

## 問3

工芸品の担当と期間の一覧を  
プログラムで完成する



# 問3

お知らせするメールの文章(図2:各工芸品の担当と期間を一覧にしたメールの文面)

工芸品 1 ... 部員 1 : 1日目～4日目  
 工芸品 2 ... 部員 2 : 1日目～1日目  
 工芸品 3 ... 部員 3 : 1日目～3日目  
 工芸品 4 ... 部員 2 : 2日目～2日目  
 工芸品 5 ... 部員 2 : 3日目～5日目

工芸品 9 ... 部員 1 : 7日目～9日目

作る工芸品(表1:各工芸品の制作日数)

工芸品	1	2	3	4	5	6	7	8	9
制作日数	4	1	3	1	3	4	2	4	3

誰が何をいつ作るのか(図1:割り当て表ー工芸品4まで)

日付(日目)	1	2	3	4	5	6	7	8	9	10	...
部員1		(工芸品)1									
部員2	(工芸品)2	(工芸品)4									
部員3		(工芸品)3									

部員は3人だよ

次の空きはわかるようになったけれど、お知らせする図2の担当と制作期間の一覧を作るのは

やっぱりめんどうですね。。。そこで自動的に担当と期間を表示できるように「プログラミング」していきます！



# 使用する配列と変数

あらたに配列「Nissu」と変数「kugeihinsu」「kougeihin」を追加するよ。

## 配列(リスト)

Nissu

工芸品の制作日数を入れる。添字は工芸品番号

Akibi

部員の次の空き日を入れる。添字は部員番号

## 変数

kugeihinsu

工芸品の総数(全部でいくつか)

kougeihin

工芸品番号を指定するために使う。例) Nissuのkougeihin番号番目

buinsu

部員の総数(全部で何人か)

buin

部員番号を指定するために使う。例) Akibiのbuin番号番目

tantou

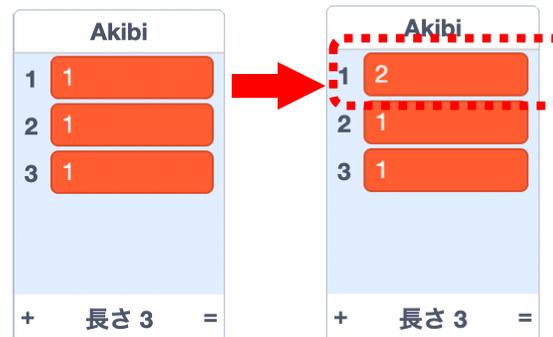
担当部員の番号を指定するために使う。Akibiのtantou番号番目



# 共通テストを解くために

配列(スクラッチの場合リスト)の使い方

Akibi ▾ の 1 番目を 2 で置き換える

$$\text{Akibi}[1] = 2$$


Akibi ▾ の tantou 番目を Akibi ▾ の tantou 番目 + Nissu ▾ の kougeihin 番目 で置き換える

$$\text{Akibi}[\text{tantou}] = \text{Akibi}[\text{tantou}] + \text{Nissu}[\text{kougeihin}]$$


### 問3 この考え方でプログラムを作ろう。

- ①配列「Nissu」に4,1,3,1,3,4,2,4,3を入れる(工芸品の製作日数)
- ②変数「kougeihinsu」に9を入れる
- ③配列「Akibi」に1,1,1を入れる(1日目から始める)
- ④変数「buinsu」に3を入れる
- ⑤変数「kougeihin」に1を入れる
- ⑥次に空いている部員をしらべる

- ① 変数「tantou」に1を入れる  
変数「buin」に2を入れる → 最初は2から始めるため
- ② [Akibi]の要素(データ)と変数を使って次に空きになる担当部員を順番に調べる  
→ 変数「tantou」に次に空きになる部員番号(「buin」)を代入する
- ③ 変数「buin」を2から1ずつ増やして、「buinsu」より大きくなるまで②を繰り返す

問2のプログラム  
次の空き部員を  
調べて  
「tantou」に入れる

- ⑦”工芸品「kougeihin」… 部員「tantou」 : 『「Akibi」の「tantou」番目』日目 ~ <?>日目“と表示する
- ⑧配列『「Akibi」の「tantou」番目』を次の空き日で置き換える
- ⑨変数「?」を1から1ずつ増やして、変数「??」まで⑥⑦⑧を繰り返す ← 問2も思い出してね



スクラッチでやってみよう！

# プログラムで考えよう

- ①配列「Nissu」に4,1,3,1,3,4,2,4,3を入れる(工芸品の製作日数)
- ②変数「kougeihinsu」に9を入れる
- ③配列「Akibi」に1,1,1を入れる(1日目から始める)
- ④変数「buinsu」に3を入れる
- ⑤変数「kougeihin」に1を入れる
- ⑥次に空いている部員をしらべる

- ① 変数「tantou」に1を入れる  
変数「buin」に2を入れる → 最初は2から始めるため
- ② [Akibi]の要素(データ)と変数を使って次に空きになる担当部員を順番に調べる  
→変数「tantou」に次に空きになる部員番号(「buin」)を代入する
- ③ 変数「buin」を2から1ずつ増やして、「buinsu」より大きくなるまで②を繰り返す

問2のプログラム



# 続きを考えよう

制作を始める日

制作を終える日

⑦”工芸品「kougeihin」… 部員「tantou」 : 『「Akibi」の「tantou」番目』日目 ~ <?>日目“と表示する

## 工芸品1の場合で考えてみると…

工芸品 1 ... 部員 1 : 1日目~4日目

工芸品1...部員1 : 1日目~4日目

日付 (日目)	1	2	3	4	5
部員1	工芸品 1				
部員2					
部員3					

	Akibi	Nissu
1	1	4
2	1	1
3	1	3
4		1



# プログラムで考えよう

制作を始める日

制作を終える日

⑦”工芸品「kougeihin」… 部員「tantou」：『「Akibi」の「tantou」番目』日目～<?>日目“と表示する

日付 (日目)	1	2	3	4	5
部員1		工芸品1			
部員2					
部員3					

Akibi		Nissu	
1	1	1	4
2	1	2	1
3	1	3	3
		4	1

工芸品の制作を終える日は、どう計算をすればよいかな？

Akibi の tantou 番目 と 日目～ と ? と 日目

工芸品 1 ... 部員 1 : 1日目～4日目

「空き」+「制作日数」だと、次の「空き」になるね！制作を終える日は1日前だから-1すればいいね！  
工芸品1の例)  $1 + 4 = 5$        $1 + 4 - 1 = 4$

# プログラムで考えよう

スクラッチでやってみよう！

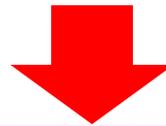
⑦”工芸品「kougihin」… 部員「tantou」 : 『「Akibi」の「tantou」番目』日目～<?>日目“と表示する

Akibi ▾ の tantou 番目

作業を始める日は  
「Akibiのtantou番目」

Akibi ▾ の tantou 番目 + Nissu ▾ の kougihin 番目 - 1

作業が終わる日は  
「Akibiのtantou番目+Nissuのkougihin番目-1」



工芸品 と kougihin と ... と 部員 と tantou と : と Akibi ▾ の tantou 番目 と 日目～ と Akibi ▾ の tantou 番目 + Nissu ▾ の kougihin 番目 - 1 と 日目 と 3 秒言う



# プログラムで考えよう

スクラッチでやってみよう！

⑦”工芸品「kougeihin」… 部員「tantou」：『「Akibi」の「tantou」番目』日目～<?>日目“と表示する

The image shows a Scratch script with the following blocks:

- tantou を 1 にする
- buin を 2 にする
- ループ: buin > buinsu まで繰り返す
- もし: Akibi の buin 番目 < Akibi の tantou 番目 なら
- tantou を buin にする
- ループ: buin を 1 ずつ変える
- ↑
- 表示: 工芸品 と kougeihin と ... と 部員 と tantou と : と Akibi の tantou 番目 と 日目~ と Akibi の tantou 番目 + Nissu の kougeihin 番目 - 1 と 日目 と 3 秒言う



# 問3

## 出題

誰が何をいつ作るのか (図5: 各工芸品の担当と期間の一覧を表示するプログラム)

Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]

kougeihinsu = 9

Akibi = [1, 1, 1]

buinsu = 3

変数? を 1 から 変数?? まで1ずつ増やしながら繰り返す:

tantou = 1

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし Akibi[buin] < Akibi[tantou]ならば:

tantou = buin

表示する(”工芸品”, kougeihin, “...”,

”部員”, tantou, “:”,

Akibi[tantou], “日目~”,

Akibi[tantou] + ? “日目”)

Akibi[tantou] = Akibi[tantou] + ?

工芸品 1 … 部員 1 : 1日目~4日目

このように表示するには ? を  
どのように指定するのがよいでしょうか?

Nissu の kougeihin 番目

Nissu[kougeihin]

Nissu の kougeihin 番目 - 1

Nissu[kougeihin] - 1

Nissu の kougeihin - 1 番目

Nissu[kougeihin - 1]

Nissu の buin 番目

Nissu[buin]

Nissu の buin 番目 - 1

Nissu[buin] - 1

Nissu の buin - 1 番目

Nissu[buin - 1]



# 問3

答え合わせ

誰が何をいつ作るのか(図5:各工芸品の担当と期間の一覧を表示するプログラム)

Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]

kougeihinsu = 9

Akibi = [1, 1, 1]

buinsu = 3

変数? を 1 から 変数?? まで1ずつ増やしながら繰り返す:

tantou = 1

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし Akibi[buin] < Akibi[tantou]ならば:

tantou = buin

表示する(”工芸品”, kougeihin, “...”,

”部員”, tantou, “ :”,

Akibi[tantou], “日目~”,

Akibi[tantou] + ? , “日目“)

Akibi[tantou] = Akibi[tantou] + ?

工芸品 1 ... 部員 1 : 1日目~4日目

空きに制作日数を足し算すると、次の空きになるから -1 すると 制作を終わる日になるね!

Nissu の kougeihin 番目 - 1 Nissu[kougeihin] - 1



# プログラムで考えよう

⑧配列『「Akibi」の「tantou」番目』を次の空き日で置き換える

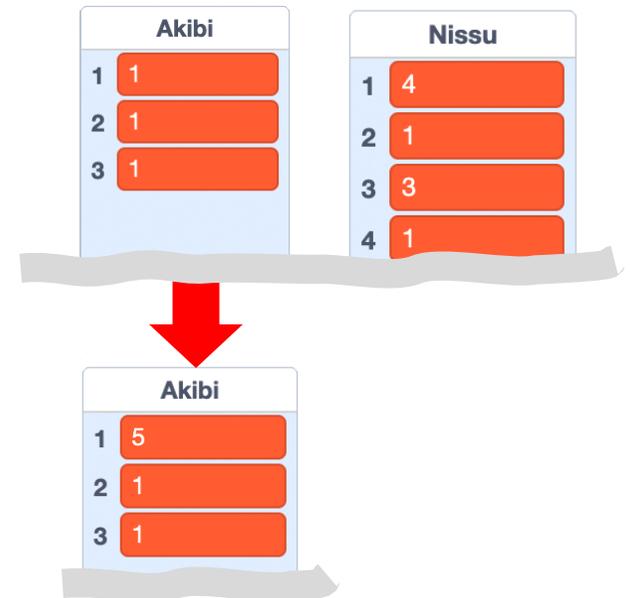
部員1が、工芸品1を作って「空き」になる日で考えると…

「Akibi」の1番目+「Nissu」の1番目でわかるね！

$$1+4=5$$

「Akibi」の1番目を5で置き換えればいいね。

日付 (日目)	1	2	3	4	5
部員1		工芸品1			
部員2					
部員3					

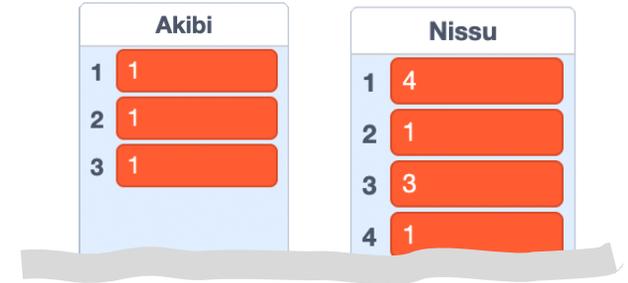


# プログラムで考えよう

スクラッチでやってみよう！

⑧配列『「Akibi」の「tantou」番目』を次の空き日で置き換える

日付 (日目)	1	2	3	4	5
部員1		工芸品1			
部員2					
部員3					



次の空きの計算を作ろう。



「空き」+「制作日数」が 次の「空き」になるね！



# プログラムで考えよう

スクラッチでやってみよう！

⑧配列『「Akibi」の「tantou」番目』を次の空き日で置き換える

日付（日目）	1	2	3	4	5
部員1		工芸品1			
部員2					
部員3					

Akibi		Nissu	
1	1	1	4
2	1	2	1
3	1	3	3
		4	1

『「Akibi」の「tantou」番目』を次の空きで置き換えよう。

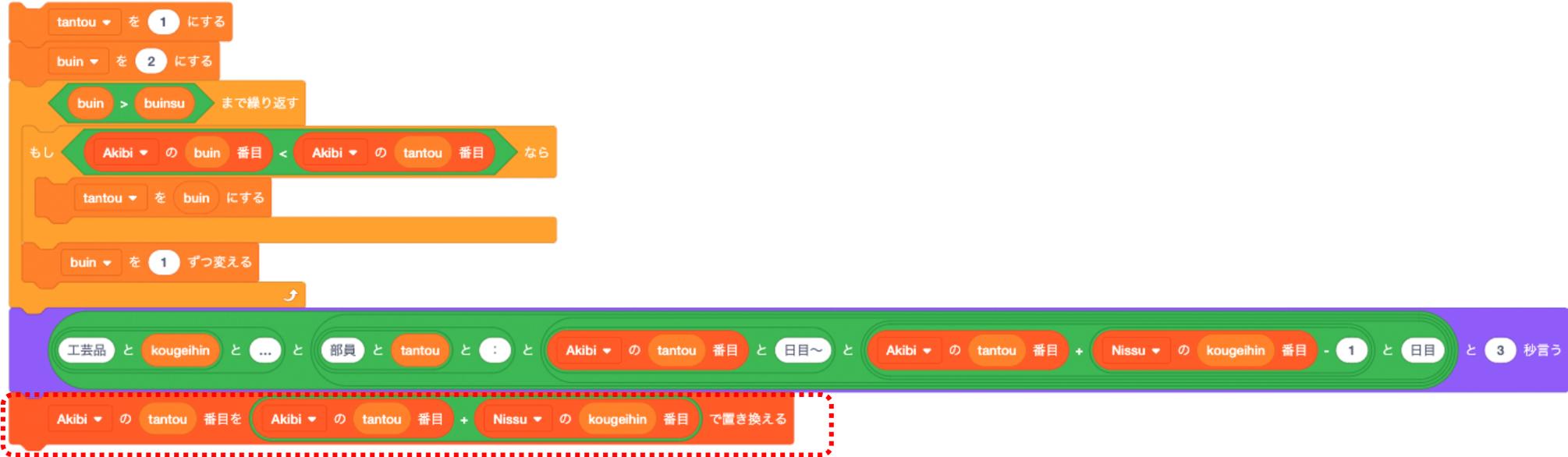
Akibi の tantou 番目を Akibi の tantou 番目 + Nissu の kougeihin 番目で置き換える



# プログラムで考えよう

スクラッチでやってみよう！

⑧配列『「Akibi」の「tantou」番目』を次の空き日で置き換える



The image shows a Scratch script designed to replace an element in an array. The script consists of the following blocks:

- tantou を 1 にする
- buin を 2 にする
- buin > buinsu まで繰り返す
- もし Akibi の buin 番目 < Akibi の tantou 番目 なら
- tantou を buin にする
- buin を 1 ずつ変える
- 
- 工芸品 と kougeihin と ... と 部員 と tantou と : と Akibi の tantou 番目 と 日目~ と Akibi の tantou 番目 + Nissu の kougeihin 番目 - 1 と 日目 と 3 秒言う
- Akibi の tantou 番目を Akibi の tantou 番目 + Nissu の kougeihin 番目 で置き換える

The final two lines of the script are enclosed in a red dashed border, indicating the specific replacement operation.



# 問3

## 出題

誰が何をいつ作るのか(図5:各工芸品の担当と期間の一覧を表示するプログラム)

```
Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
kougeihinsu = 9
Akibi = [1, 1, 1]
buinsu = 3
```

変数? を 1 から 変数?? まで1ずつ増やしながら繰り返す:

```
tantou = 1
```

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし Akibi[buin] < Akibi[tantou]ならば:

```
tantou = buin
```

表示する(”工芸品”, kougeihin, “...”,

”部員”, tantou, “:”,

Akibi[tantou], “日目~”,

Akibi[tantou] + [?] , “日目“)

```
Akibi[tantou] = Akibi[tantou] + [?]
```

配列『「Akibi」の「tantou」番目』を次の空き日で置き換えるには [?] に入れるのは?

Nissu の kougeihin 番目

Nissu[kougeihin]

Nissu の kougeihin 番目 - 1

Nissu[kougeihin] - 1

Nissu の kougeihin - 1 番目

Nissu[kougeihin - 1]

Nissu の buin 番目

Nissu[buin]

Nissu の buin 番目 - 1

Nissu[buin] - 1

Nissu の buin - 1 番目

Nissu[buin - 1]



# 問3

答え合わせ

誰が何をいつ作るのか(図5:各工芸品の担当と期間の一覧を表示するプログラム)

```
Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
kougeihinsu = 9
Akibi = [1, 1, 1]
buinsu = 3
```

変数? を 1 から 変数?? まで1ずつ増やしながら繰り返す:

```
tantou = 1
```

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし Akibi[buin] < Akibi[tantou]ならば:

```
tantou = buin
```

表示する(”工芸品”, kougeihin, “...”,

”部員”, tantou, “:”,

Akibi[tantou], “日目~”,

Akibi[tantou] + [?], “日目“)

```
Akibi[tantou] = Akibi[tantou] + [?]
```

配列『「Akibi」の「tantou」番目』を次の空き日で置き換えるには 制作日数を足し算すればよいよね！

Nissu の kougeihin 番目

Nissu[kougeihin]



# プログラムで考えよう

- ⑥ 問2のプログラム(次の空きを「tantou」に入れる)
- ⑦ ”工芸品「kougeihin」…部員「tantou」 : 『「Akibi」の「tantou」番目』日目 ~<?>日目“と表示する
- ⑧ 配列『「Akibi」の「tantou」番目』を次の空き日で置き換える

⑨ 変数「 ? 」を1から1ずつ増やして、「 ?? 」まで⑥⑦⑧を繰り返す

工芸品を1から9まで担当を決めたいよね。「 ? 」と「 ?? 」にはいる変数はなんだろう？

⑥⑦⑧ で kougeihin の担当を決めているよね！

ということは「kougeihin」(最初は1)から「kougeihinsu」(工芸品の総数 9)まで、「kougeihin」を1ずつ増やしながらか繰り返すといいよね！



# プログラムで考えよう

- ⑥ 問2のプログラム(次の空きを「tantou」に入れる)
- ⑦ ”工芸品「kougeihin」…部員「tantou」：『「Akibi」の「tantou」番目』日目～<?>日目“と表示する
- ⑧ 配列『「Akibi」の「tantou」番目』を次の空き日で置き換える

⑨ 変数「kougeihin」を1から1ずつ増やして、「kougeihinsu」まで⑥⑦⑧を繰り返す

変数「kougeihin」を1増やすには、スクラッチは次のブロックを使うんだね！



変数「kougeihinsu」(9が代入されている)まで繰り返し実行する条件はこのブロックだよ。



# プログラムで考えよう

## ⑨変数「kougeihin」を1から1ずつ増やす

```
tantou を 1 にする
buin を 2 にする
ループ: buin > buinsu まで繰り返す
  もし Akibi の buin 番目 < Akibi の tantou 番目 なら
    tantou を buin にする
  buin を 1 ずつ変える
ループ終了
[ ] 工艺品 と kougeihin と ... と [ ] 部員 と tantou と [ ] と Akibi の tantou 番目 と 日目～
Akibi の tantou 番目を Akibi の tantou 番目 + Nissu の kougeihin 番目 で置き換える
kougeihin を 1 ずつ変える
```

スクラッチでやってみよう!

```
定義 問3
Nissu のすべてを削除する
4 を Nissu に追加する
1 を Nissu に追加する
3 を Nissu に追加する
1 を Nissu に追加する
3 を Nissu に追加する
4 を Nissu に追加する
2 を Nissu に追加する
4 を Nissu に追加する
3 を Nissu に追加する
kougeihinsu を 9 にする
Akibi のすべてを削除する
1 を Akibi に追加する
1 を Akibi に追加する
1 を Akibi に追加する
buinsu を 3 にする
kougeihin を 1 にする
```



# プログラムで考えよう

スクラッチでやってみよう！

⑨変数「kougeihin」を1から1ずつ増やして、「kougeihinsu」まで⑥⑦⑧を繰り返す

A Scratch script enclosed in a red dashed box. It starts with a loop block: "kougeihin > kougeihinsu まで繰り返す". Inside the loop, it sets "tantou" to 1, then "buin" to 2. Another loop block follows: "buin > buinsu まで繰り返す". Inside this loop, there is an "if" block: "もし Akibi の buin 番目 < Akibi の tantou 番目 なら". The "if" block contains: "tantou を buin にする", a blank block, and "buin を 1 ずつ変える". Below the "if" block is a "say" block: "工艺品 と kougeihin と ... と 部員 と tantou と : と Akibi の tantou 番目 と 日目〜 と". Below the "say" block is: "Akibi の tantou 番目を Akibi の tantou 番目 + Nissu の kougeihin 番目 で置き換える", and finally "kougeihin を 1 ずつ変える".

A vertical stack of Scratch code blocks. From top to bottom: "1 を Nissu に追加する", "3 を Nissu に追加する", "4 を Nissu に追加する", "2 を Nissu に追加する", "4 を Nissu に追加する", "3 を Nissu に追加する", "kougeihinsu を 9 にする" (highlighted with a red dashed box), "Akibi のすべてを削除する", "1 を Akibi に追加する", "1 を Akibi に追加する", "1 を Akibi に追加する", "buinsu を 3 にする", and "kougeihin を 1 にする" (highlighted with a red dashed box).



# 問3

## 出題

誰が何をいつ作るのか (図5: 各工芸品の担当と期間の一覧を表示するプログラム)

```
Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
```

```
kougeihinsu = 9
```

```
Akibi = [1, 1, 1]
```

```
buinsu = 3
```

変数? を 1 から 変数?? まで1ずつ増やしながら繰り返す:

```
tantou = 1
```

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし Akibi[buin] < Akibi[tantou]ならば:

```
tantou = buin
```

表示する(”工芸品”, kougeihin, “...”,

”部員”, tantou, “:”,

Akibi[tantou], “日目~”,

Akibi[tantou] + ? , “日目“)

```
Akibi[tantou] = Akibi[tantou] + ?
```

変数? 変数?? に入れるのは?

buin

buinsu

kougeihin

kougeihinsu

tantou



# 問3

答え合わせ

誰が何をいつ作るのか (図5: 各工芸品の担当と期間の一覧を表示するプログラム)

```
Nissu = [4, 1, 3, 1, 3, 4, 2, 4, 3]
kougeihinsu = 9
Akibi = [1, 1, 1]
buinsu = 3
```

変数? を 1 から 変数?? まで1ずつ増やしながら繰り返す:

```
tantou = 1
```

buin を 2 から buinsuまで1ずつ増やしながら繰り返す:

もし Akibi[buin] < Akibi[tantou]ならば:

```
tantou = buin
```

表示する(”工芸品”, kougeihin, “...”,  
”部員”, tantou, “:”,  
Akibi[tantou], “日目~”,  
Akibi[tantou] + ? , “日目“)

```
Akibi[tantou] = Akibi[tantou] + ?
```

変数「kougeihin」を1から1ずつ増やして、  
「kougeihinsu」まで繰り返して、工芸品の担当を  
表示する(⑥⑦⑧のプログラム)んだよね!

変数?

kougeihin

変数??

kougeihinsu



# プログラムで考えよう

スクラッチでやってみよう！

定義 関数

- Nissu のすべてを削除する
- 4 を Nissu に追加する
- 1 を Nissu に追加する
- 3 を Nissu に追加する
- 1 を Nissu に追加する
- 3 を Nissu に追加する
- 4 を Nissu に追加する
- 2 を Nissu に追加する
- 4 を Nissu に追加する
- 3 を Nissu に追加する
- kougeihin を 0 にする
- Akibi のすべてを削除する
- 1 を Akibi に追加する
- 1 を Akibi に追加する
- 1 を Akibi に追加する
- buinsu を 3 にする
- kougeihin を 1 にする
- kougeihin > kougeihinu まで繰り返す
- tantou を 1 にする
- buin を 2 にする
- buin > buinsu まで繰り返す
- もし Akibi の buin 番目 < Akibi の tantou 番目 なら
  - tantou を buin にする
  - buin を 1 ずつ変える
- 工程名 と kougeihin と ... と 部員 と tantou と 1 と Akibi の tantou 番目 と 日目〜 と Akibi の tantou 番目 + Nissu の kougeihin 番目 - 1 と 日目 と 3 秒待つ
- 工程表に記録 工程名 kougeihin 部員 tantou 作業開始日 Akibi の tantou 番目 作業完了日 Akibi の tantou 番目 + Nissu の kougeihin 番目 - 1
- Akibi の tantou 番目を Akibi の tantou 番目 + Nissu の kougeihin 番目 で置き換える
- kougeihin を 1 ずつ変える



# プログラムで考えよう

スクラッチでやってみよう！

リスト「工芸」に結果を追加するために、このブロックを用意しておいたよ。  
セリフの次か前に入れてね！

工程表に記録 工芸品 **kougeihin** 部員 **tantou** 作業開始日 **Akibi** の **tantou** 番目 作業完了日 **Akibi** の **tantou** 番目 + **Nissu** の **kougeihin** 番目 - 1

工程	
1	部員1：1日目～4日目
2	部員2：1日目～1日目
3	部員3：1日目～3日目
4	部員2：2日目～2日目
5	部員2：3日目～5日目
6	部員3：4日目～7日目
7	部員1：5日目～6日目
8	部員2：6日目～9日目
9	部員1：7日目～9日目
+	長さ 9 =



スクラッチでやってみよう！

プログラム完成！！



ユ-バ- プログラミング スク-ル

準備ができたなら  
共通テストの第3問を解いてみよう！

文章は少し難しいかもしれないけれど  
しっかり読んで「共通テストを解くために」  
のところを見直せば  
きっと正解できると思うよ！



おさえおこう！

# 共通テストを解くために

## 変数の使い方

buinsu を 3 にする

tantou を buin にする

buinsu = 3

tantou = buin

## 配列(スクラッチの場合リスト)の使い方

Akibi のすべてを削除する

5 を Akibi に追加する

3 を Akibi に追加する

4 を Akibi に追加する

Akibi の 1 番目

Akibi の tantou 番目

Akibi = [5,3,4]

Akibi [1]

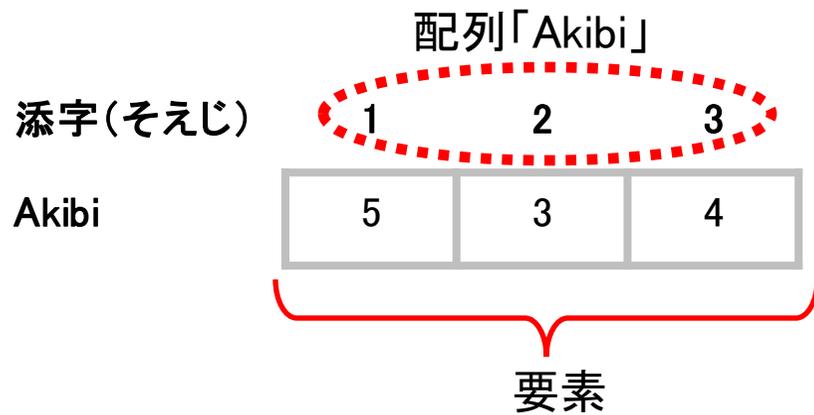
Akibi [tantou]

(そえじ)  
添字の指定



# 共通テストを解くために

## 添字(そえじ)、要素 とは



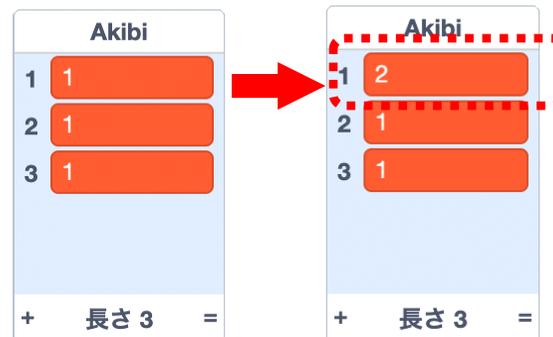
「添字」は、スクラッチのリストだと、1番目、2番目、3番目…の数字と同じようなものだよ。  
配列(リスト)には**たくさんのデータ**を入れることができるから、  
何番目のデータなのか、わかるようにするんだよね！  
**データ**のことを「要素」というよ。



# 共通テストを解くために

配列(スクラッチの場合リスト)の使い方

Akibi の 1 番目を 2 で置き換える

$$\text{Akibi}[1] = 2$$


Akibi の tantou 番目を Akibi の tantou 番目 + Nissu の kougeihin 番目で置き換える

$$\text{Akibi}[\text{tantou}] = \text{Akibi}[\text{tantou}] + \text{Nissu}[\text{kougeihin}]$$


# 共通テストを解くために

使用する配列と変数を整理しておく

配列(リスト)	
Nissu	工芸品の制作日数を入れる。添字は工芸品番号
Akibi	部員の次の空き日を入れる。添字は部員番号

変数	
kougeihinsu	工芸品の総数(全部でいくつか)
kougeihin	工芸品番号を指定するために使う。例) Nissuのkougeihin番号番目
buinsu	部員の総数(全部で何人か)
buin	部員番号を指定するために使う。例) Akibiのbuin番号番目
tantou	担当部員の番号を指定するために使う。Akibiのtantou番号番目



## この教材で使用している スクラッチ (Scratch) について

Scratch はスクラッチ財団ならびにマサチューセッツ工科大学メディアラボライフロング・キンダーガーデン・グループのプロジェクトです。詳細はこちらを参照してください。 <https://scratch.mit.edu>  
Scratchのロゴ、Scratch Cat、Gobo、Pico、Nano、Giga、Tera はScratch チームの商標です。  
マサチューセッツ工科大学 メディアラボライフロング・キンダーガーデン・グループならびにScratchチームは、この教材のスポンサーではありません。またこの教材を推奨しているものでもありません。

## 謝辞

Scratchは、スクラッチ財団ならびにマサチューセッツ工科大学 メディアラボライフロング・キンダーガーデン・グループのプロジェクトで、無償で提供されています。  
子ども達にこのような素晴らしい学習環境を提供して下さったチームの皆様、Scratchの開発に貢献されている皆様に心から感謝申し上げます。

## この教材ならびに関連する動画について

本教材 (PDF・スクラッチプロジェクトファイル・動画) は、商用・非商用を問わず自由にご活用いただけます。学校や塾、企業研修はもちろん、ご家庭での学習にもぜひお役立てください。改変・再配布も可能ですので、より多くの方に届くようご活用いただければ幸いです。

### 免責事項

本教材は現状のまま提供され、内容や動作についていかなる保証も行いません。また、本教材の使用により発生したいかなる損害 (データの消失、システムの不具合、その他のトラブルを含む) についても、当社は責任を負いません。ご利用はすべて自己責任でお願いいたします。

なお、本教材の著作権は当社に帰属します。

ユーバー株式会社  
info@yuber.jp

