

速習！情報 ～共通テスト対策講座～

プログラミング (二次元配列)

二次元配列、入れ子の反復構造

一次元配列は、配列名[添字]で個々の値にアクセスする

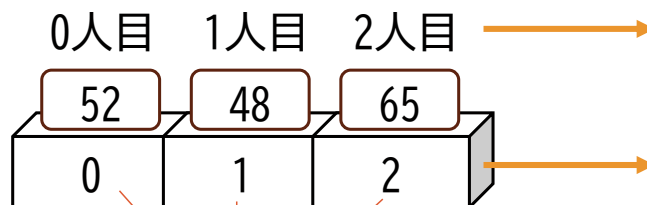
生徒	A	B	C
英語	52	48	65

列が増える

イメージ

配列名

Ten



配列 Ten = [52, 48, 65]

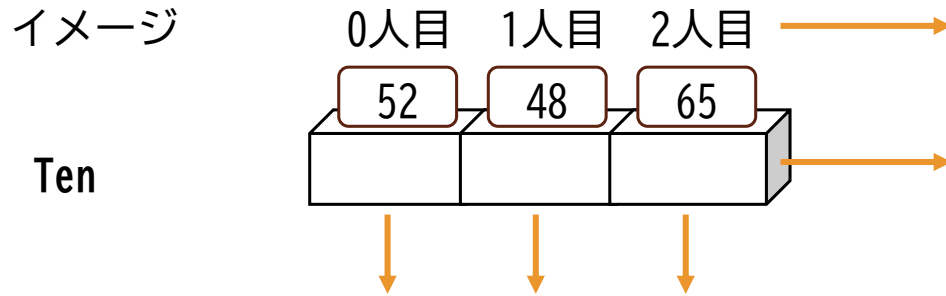
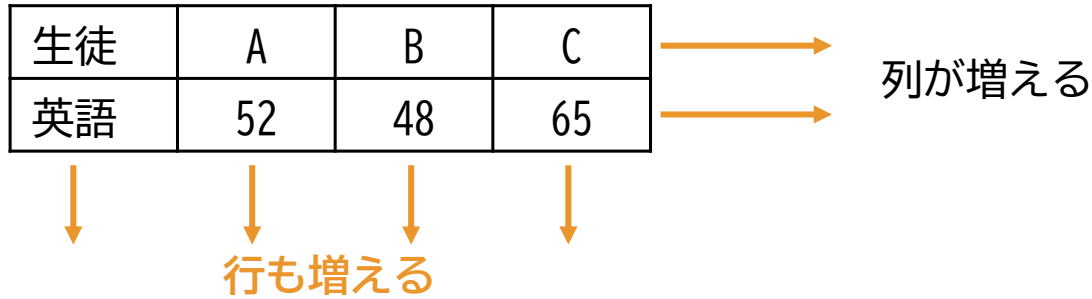
Ten[0]=52

Ten[1]=48

Ten[2]=65

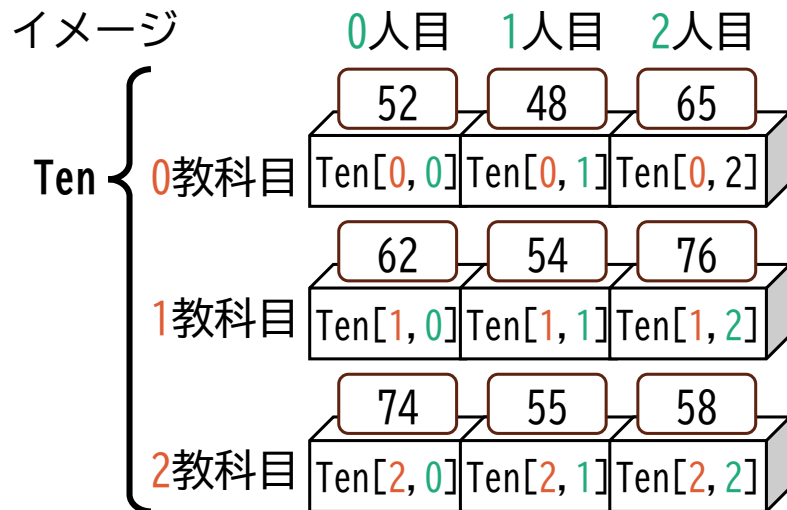
- 添字の開始が0のプログラミング言語と1のものがある
- 共通テストのDNCLでは0開始が標準だが、1開始の問題もある

二次元配列は列だけでなく行も増える



二次元配列は表データを扱う

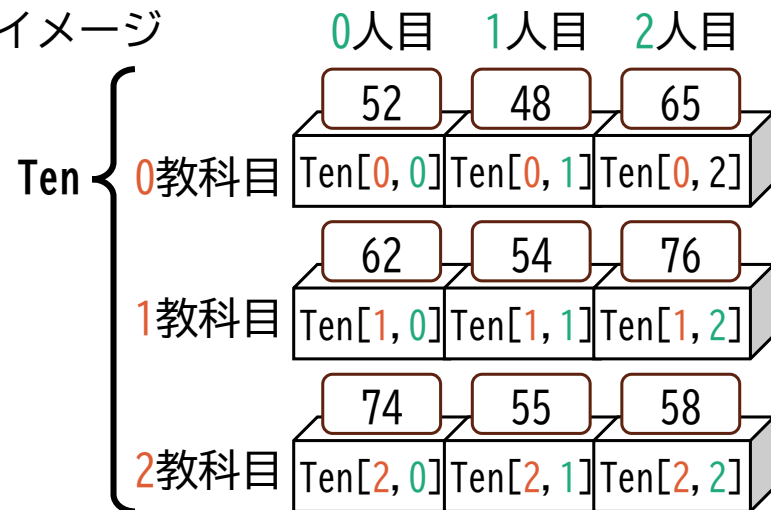
生徒	A	B	C
英語	52	48	65
数学	62	54	76
国語	74	55	58



二次元配列は表データを扱い、配列名[添字, 添字]で個々の値にアクセスする

生徒	A	B	C
英語	52	48	65
数学	62	54	76
国語	74	55	58

イメージ



Ten[0,0]=52 Ten[0,1]=48 Ten[0,2]=65

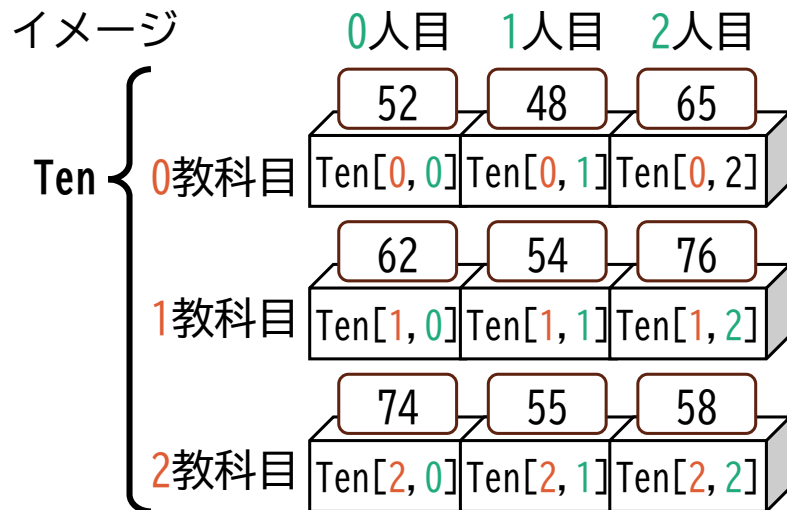
Ten[1,0]=62 Ten[1,1]=54 Ten[1,2]=76

Ten[2,0]=74 Ten[2,1]=55 Ten[2,2]=58

二次元配列は表データを扱い、配列名[添字, 添字]で個々の値にアクセスする

生徒	A	B	C
英語	52	48	65
数学	62	54	76
国語	74	55	58

※特に添字の2つの変数の意味を把握する



教科順 生徒順

Ten[0,0]=52 Ten[0,1]=48 Ten[0,2]=65

Ten[1,0]=62 Ten[1,1]=54 Ten[1,2]=76

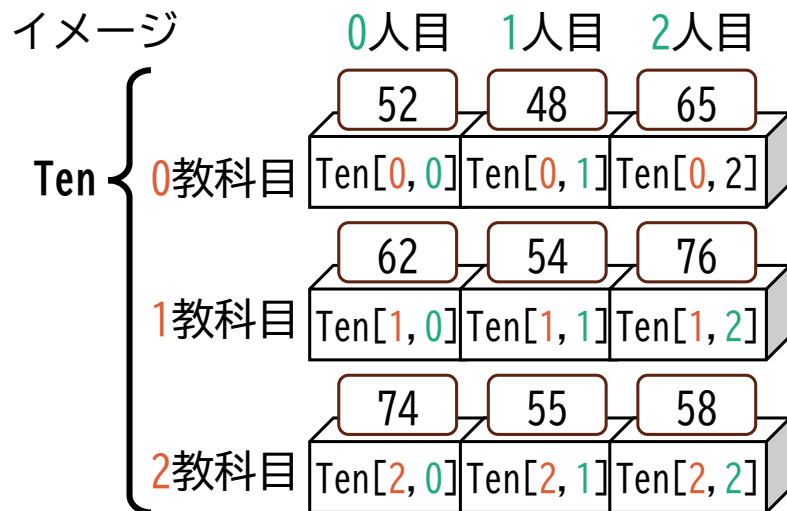
Ten[2,0]=74 Ten[2,1]=55 Ten[2,2]=58

二次元配列は表データを扱い、配列名[添字, 添字]で個々の値にアクセスする

生徒	A	B	C
英語	52	48	65
数学	62	54	76
国語	74	55	58

※特に添字の2つの変数の意味を把握する

配列 Ten = [[52, 48, 65], [62, 54, 76], [74, 55, 58]]



教科順 生徒順

Ten[0,0]=52 Ten[0,1]=48 Ten[0,2]=65

Ten[1,0]=62 Ten[1,1]=54 Ten[1,2]=76

Ten[2,0]=74 Ten[2,1]=55 Ten[2,2]=58

二次元配列は表データを扱い、配列名[添字, 添字]で個々の値にアクセスする

生徒	A	B	C
英語	52	48	65
数学	62	54	76
国語	74	55	58

※特に添字の2つの変数の意味を把握する

添字が手前の「教科」のグループ

配列 Ten = [[52, 48, 65], [62, 54, 76], [74, 55, 58]]

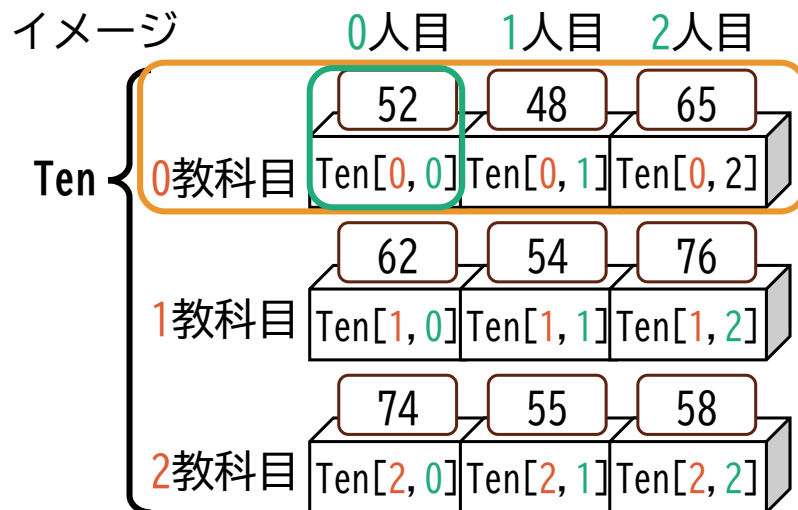
添字が後ろの「生徒」が「教科」の中に入る

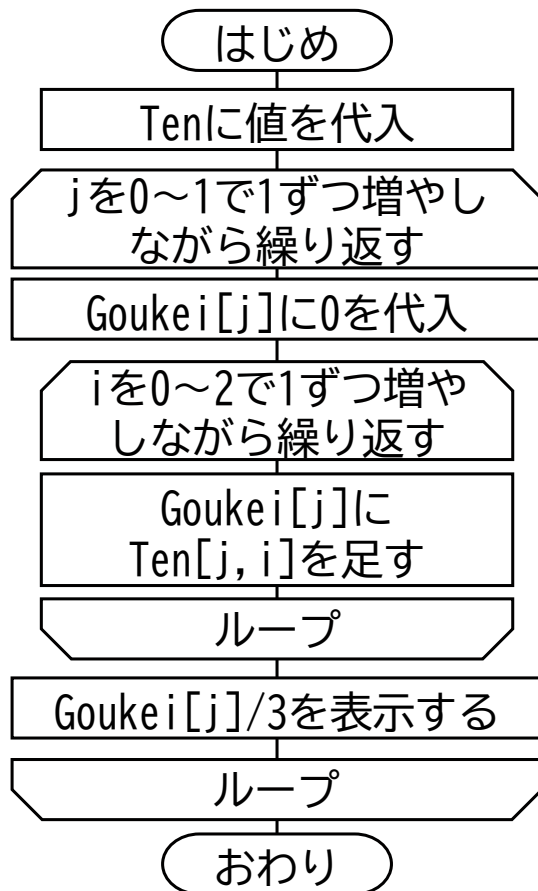
教科順 生徒順

Ten[0][0]=52 Ten[0,1]=48 Ten[0,2]=65

Ten[1,0]=62 Ten[1,1]=54 Ten[1,2]=76

Ten[2,0]=74 Ten[2,1]=55 Ten[2,2]=58





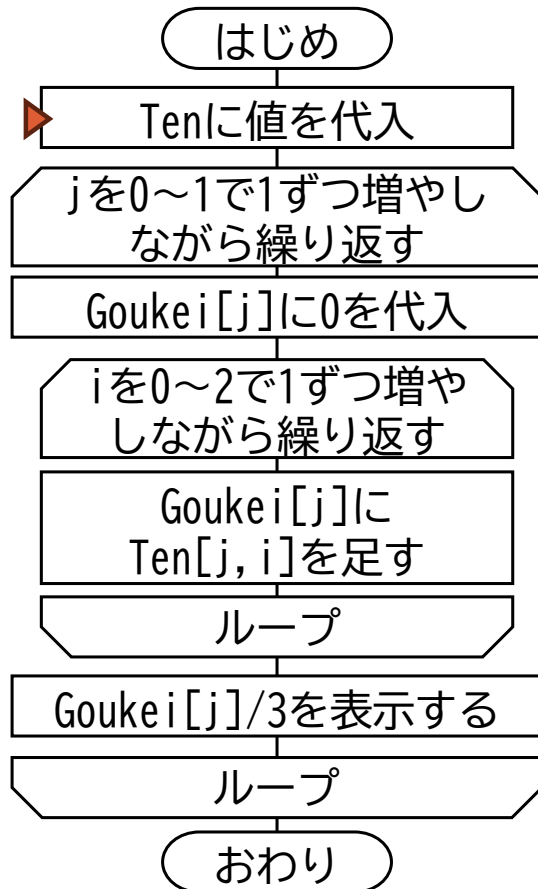
共通テストで使用されるDNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└ ┌ Goukei[j] = Goukei[j] + Ten[j, i]
  └ 表示する(Goukei[j]/3)
    
```

生徒	A	B	C
英語	52	48	65
数学	62	54	76

- 「=」は代入の意味 ※等しいではない



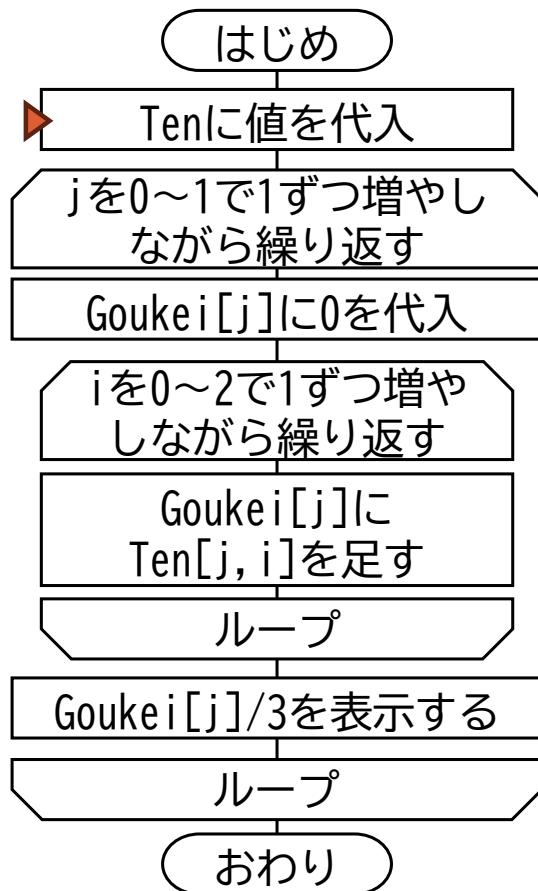
共通テストで使用されるDNCL

```

▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└ ┌ Goukei[j] = Goukei[j] + Ten[j, i]
  └ 表示する(Goukei[j]/3)
    
```

生徒	A	B	C
英語	52	48	65
数学	62	54	76

- 「=」は代入の意味 ※等しいではない

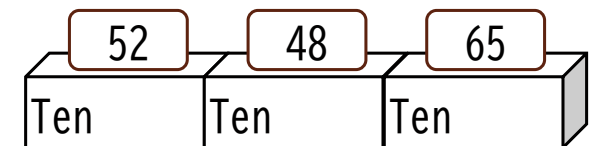


共通テストで使用されるDNCL

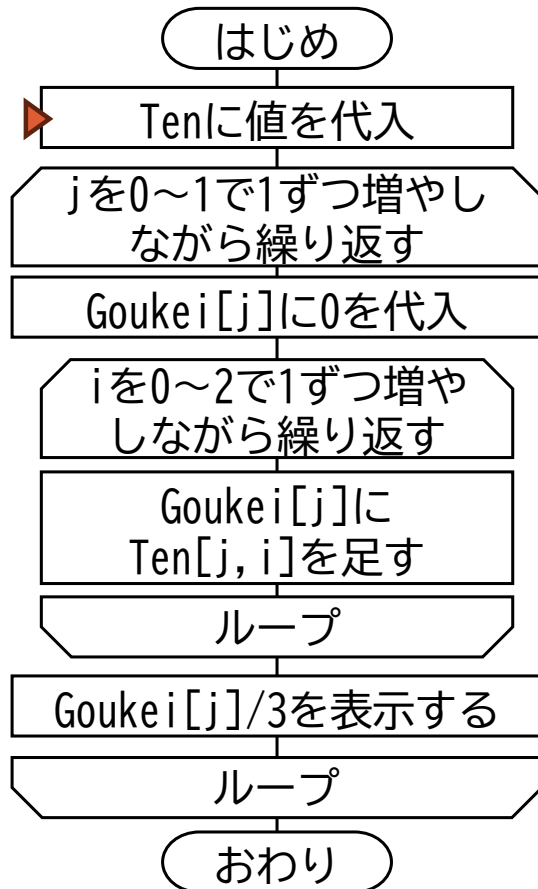
```

    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | 表示する(Goukei[j]/3)
  
```

生徒	A	B	C
英語	52	48	65
数学	62	54	76



- 「=」は代入の意味 ※等しいではない

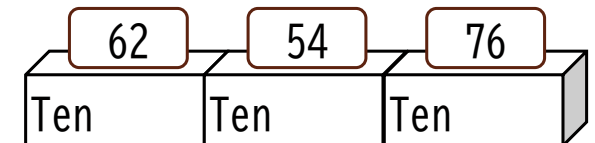
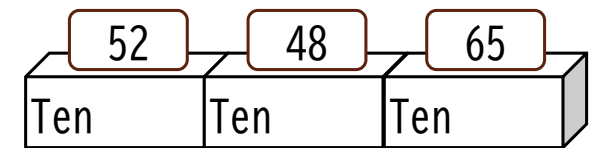


共通テストで使用されるDNCL

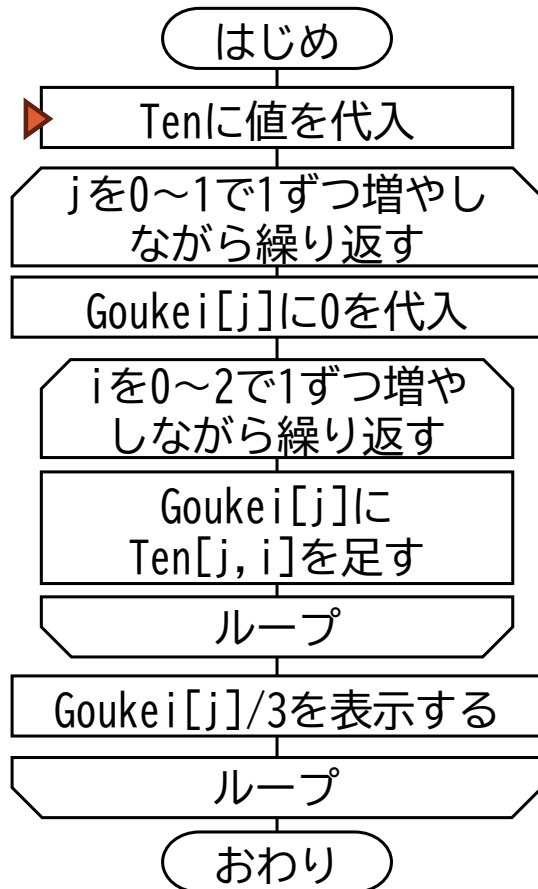
```

    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | 表示する(Goukei[j]/3)
  
```

生徒	A	B	C
英語	52	48	65
数学	62	54	76



- 「=」は代入の意味 ※等しいではない



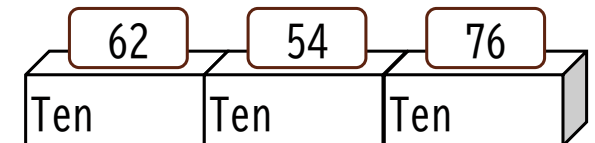
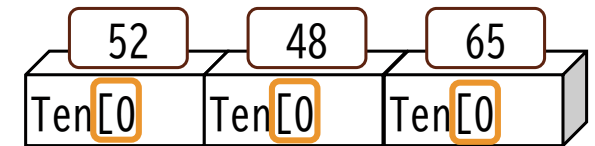
共通テストで使用されるDNCL

```

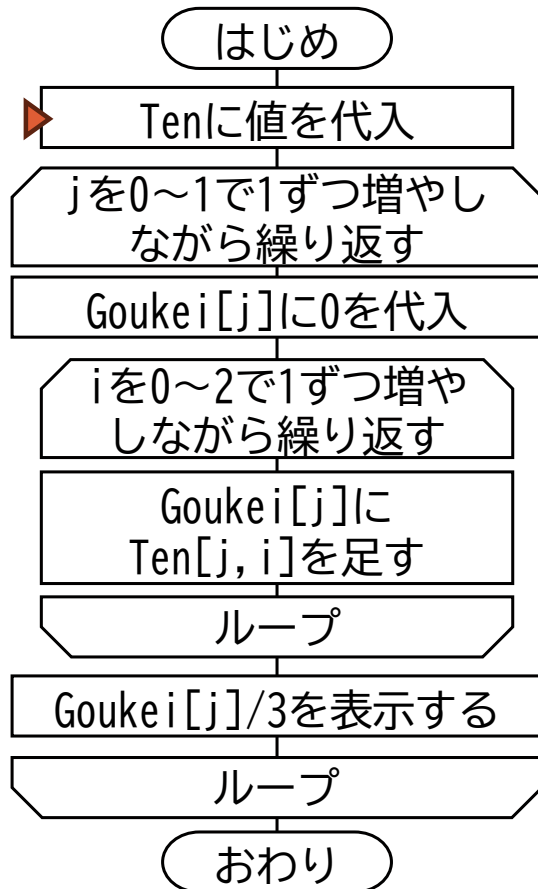
    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | 表示する(Goukei[j]/3)
  
```

生徒	A	B	C
0 英語	52	48	65
数学	62	54	76

手前の添字は行番号



- 「=」は代入の意味 ※等しいではない



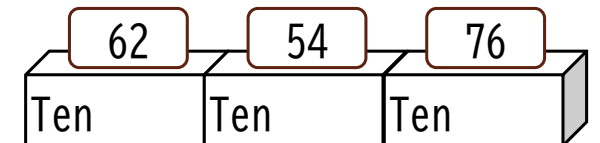
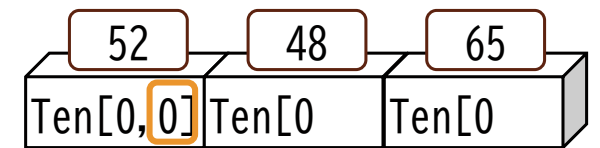
共通テストで使用されるDNCL

```

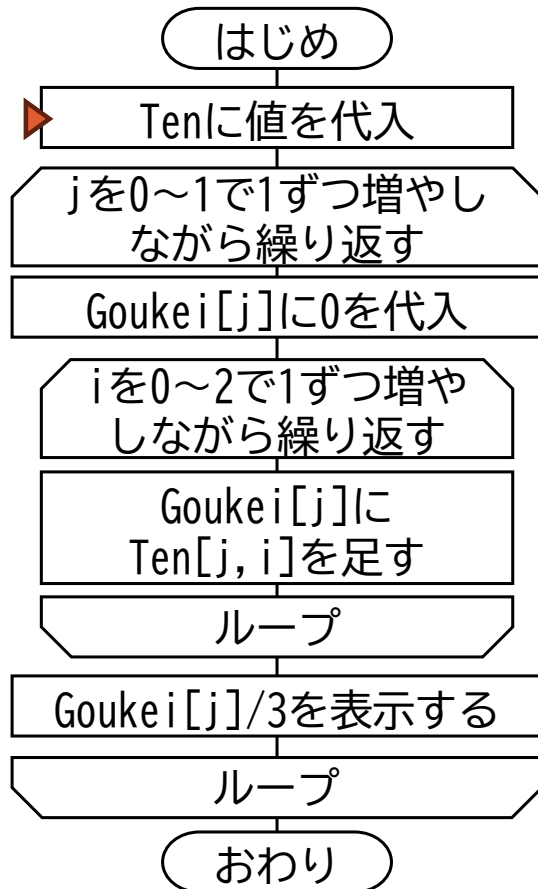
    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    |   Goukei[j] = 0
    |   iを0から2まで1ずつ増やしながら繰り返す：
    |   |   Goukei[j] = Goukei[j] + Ten[j, i]
    |   |   表示する(Goukei[j]/3)
  
```

	0			
生徒	A	B	C	
0 英語	52	48	65	
数学	62	54	76	

手前の添字は行番号、後ろの添字は列番号



- 「=」は代入の意味 ※等しいではない



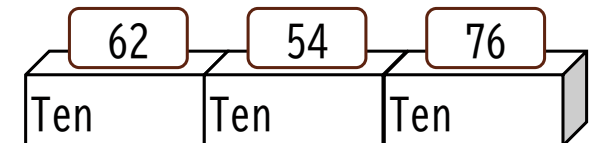
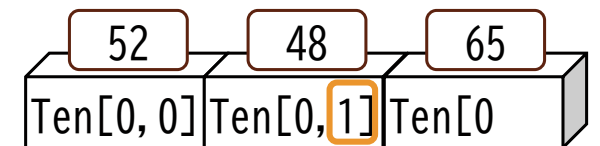
共通テストで 사용되는 DNCL

```

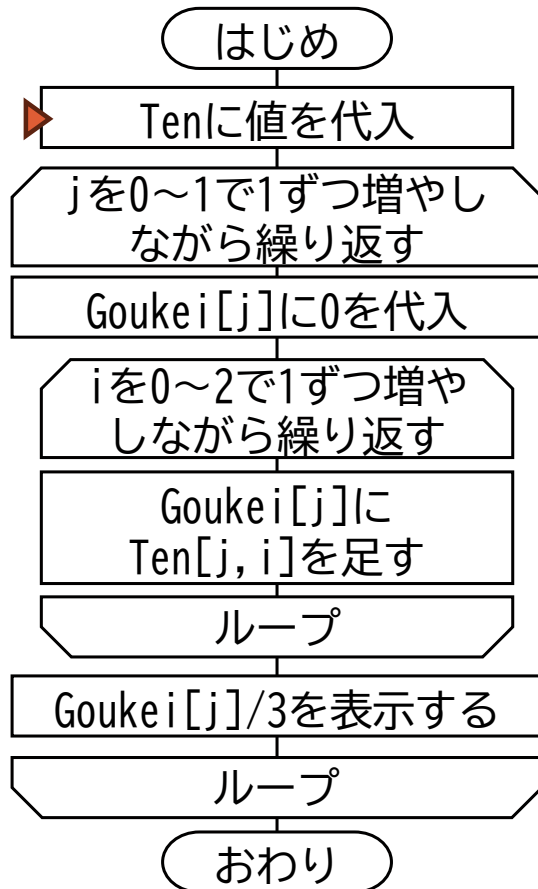
    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | 表示する(Goukei[j]/3)
  
```

	0	1	
生徒	A	B	C
英語	52	48	65
数学	62	54	76

手前の添字は行番号、後ろの添字は列番号



- 「=」は代入の意味 ※等しいではない



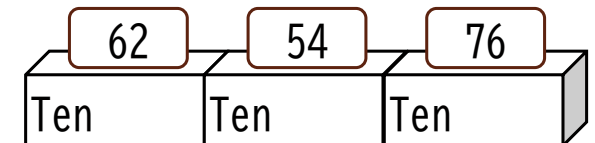
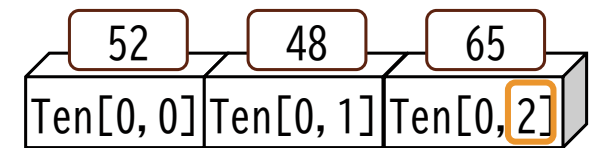
共通テストで 사용되는DNCL

```

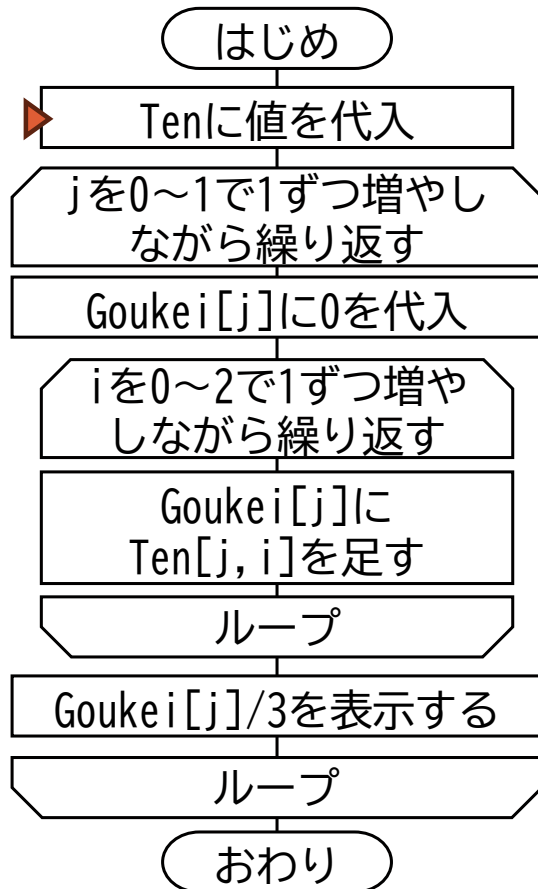
    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | 表示する(Goukei[j]/3)
  
```

	0	1	2
生徒	A	B	C
英語	52	48	65
数学	62	54	76

手前の添字は行番号、後ろの添字は列番号



- 「=」は代入の意味 ※等しいではない



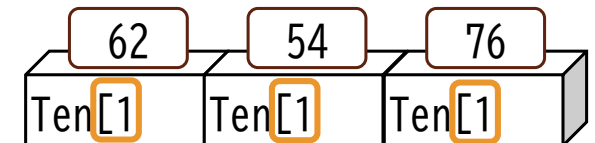
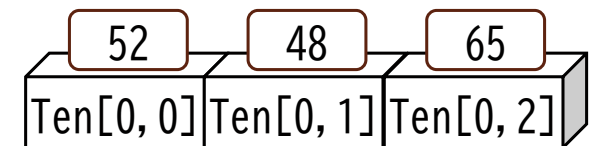
共通テストで使用されるDNCL

```

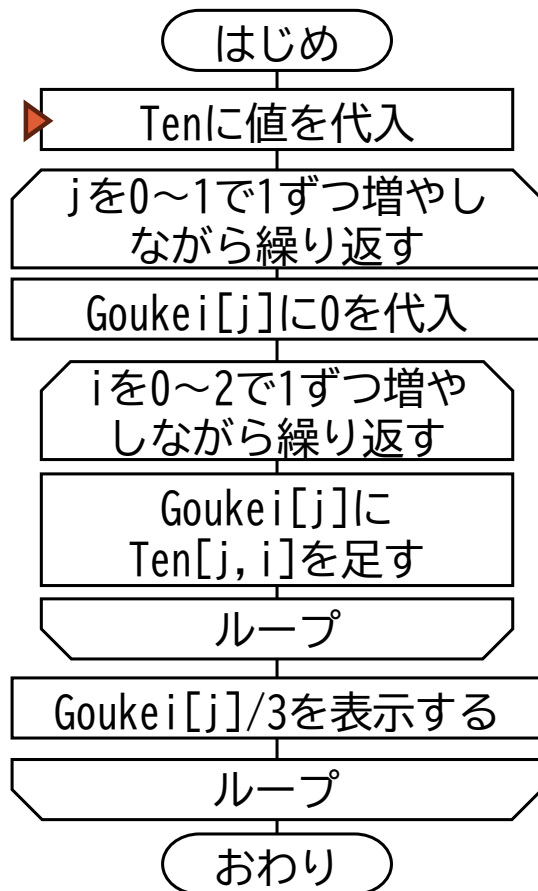
    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | 表示する(Goukei[j]/3)
  
```

		0	1	2
生徒		A	B	C
0 英語		52	48	65
1 数学		62	54	76

手前の添字は行番号、後ろの添字は列番号



- 「=」は代入の意味 ※等しいではない



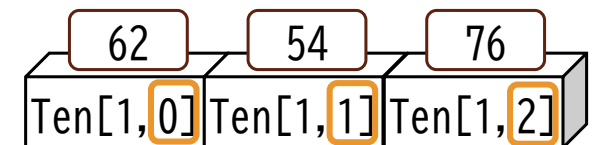
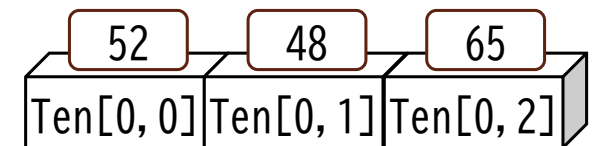
共通テストで使用されるDNCL

```

    ▶ Ten = [ [52, 48, 65], [62, 54, 76] ]
    jを0から1まで1ずつ増やしながら繰り返す：
    | Goukei[j] = 0
    | iを0から2まで1ずつ増やしながら繰り返す：
    | | Goukei[j] = Goukei[j] + Ten[j, i]
    | | 表示する(Goukei[j]/3)
  
```

	0	1	2
生徒	A	B	C
0 英語	52	48	65
1 数学	62	54	76

手前の添字は行番号、後ろの添字は列番号



変数の意味の把握

- 「=」は代入の意味 ※等しいではない

共通テストで使用されるDNCL

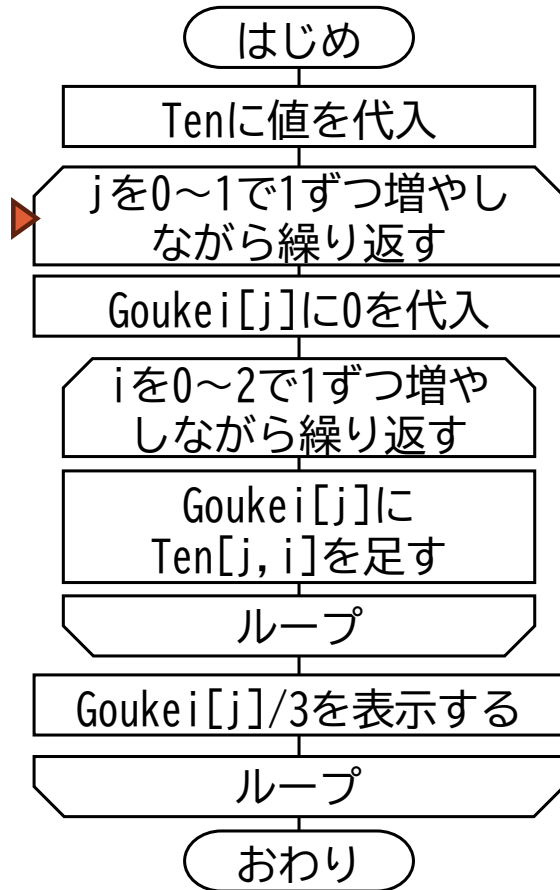
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j][i]
  表示する(Goukei[j]/3)

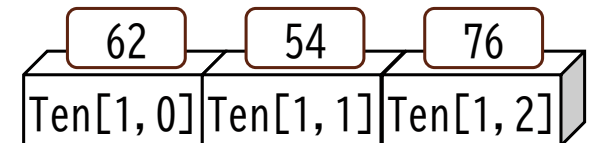
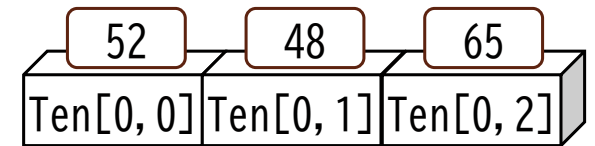
```

※変数の意味を把握する

行番号なので、
教科順を示す

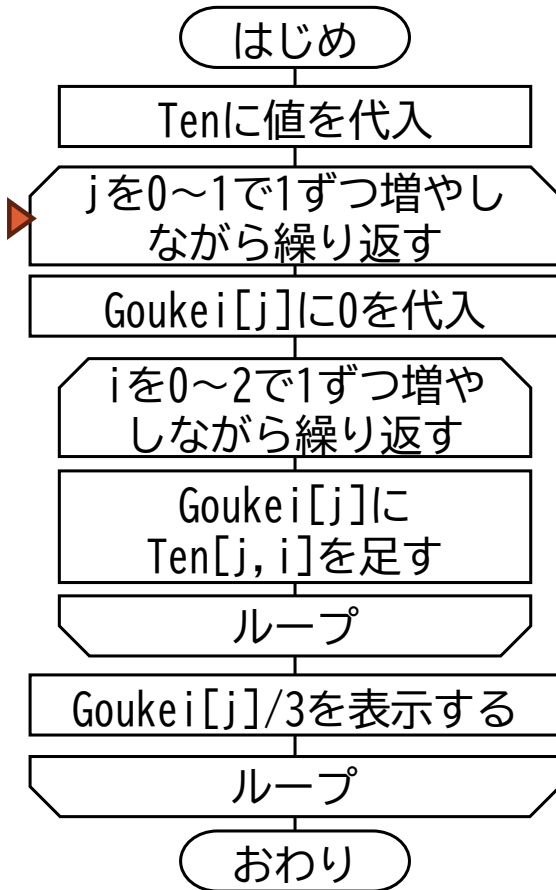


	0	1	2
生徒	A	B	C
0 英語	52	48	65
1 数学	62	54	76



変数の意味の把握

- 「=」は代入の意味 ※等しいではない



共通テストで使用されるDNCL

```

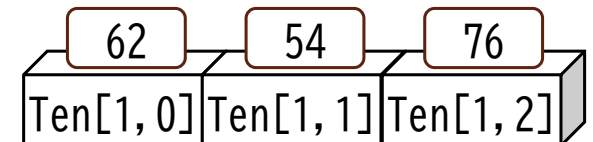
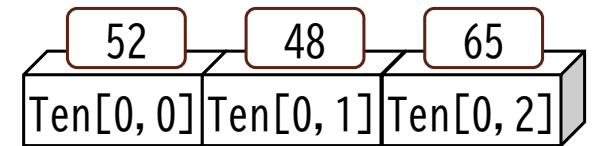
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

行番号なので、教科順を示す

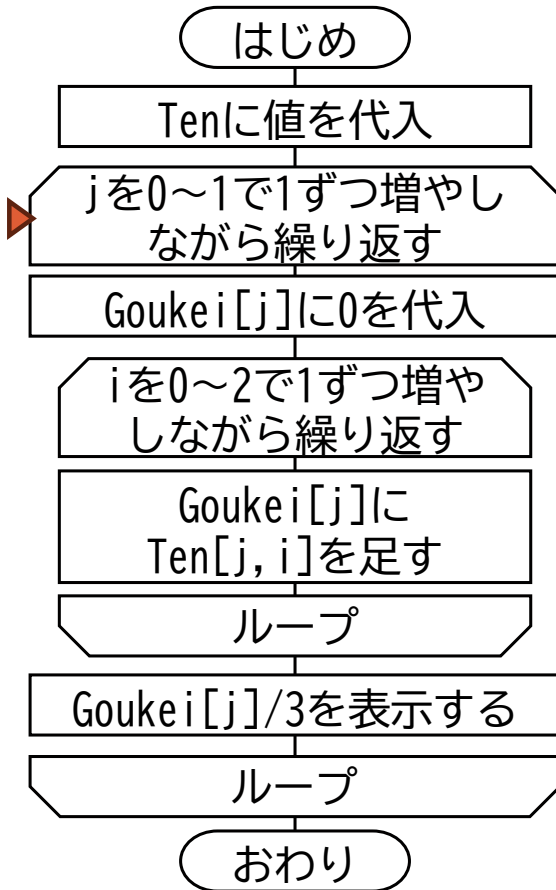
	0	1	2
生徒	A	B	C
英語	52	48	65
数学	62	54	76

j=0
j=1



変数の意味の把握

- 「=」は代入の意味 ※等しいではない



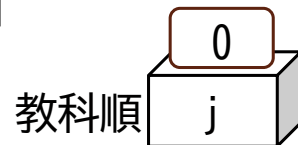
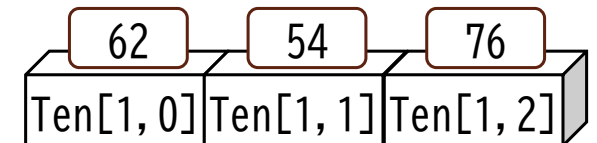
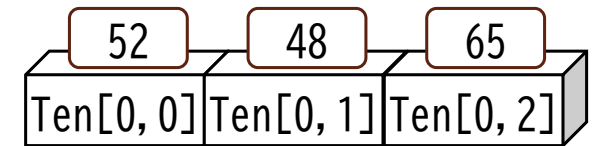
共通テストで 사용되는DNCL

```

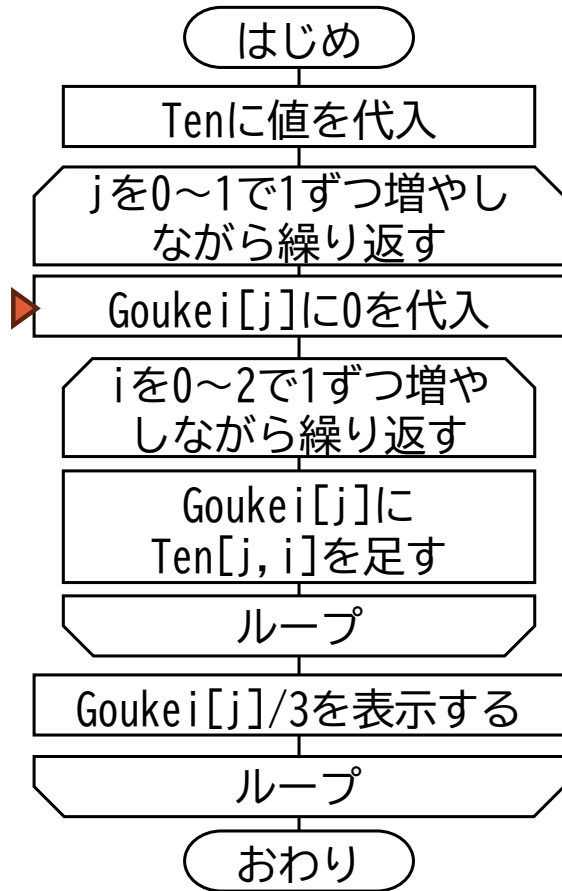
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

	0	1	2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握



共通テストで使用されるDNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す :
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す :
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

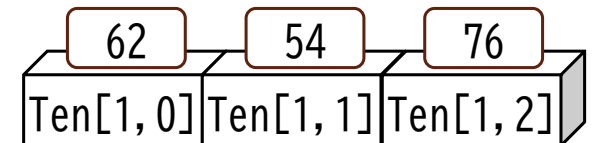
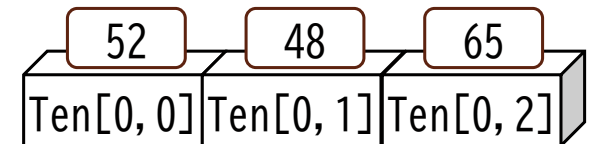
※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

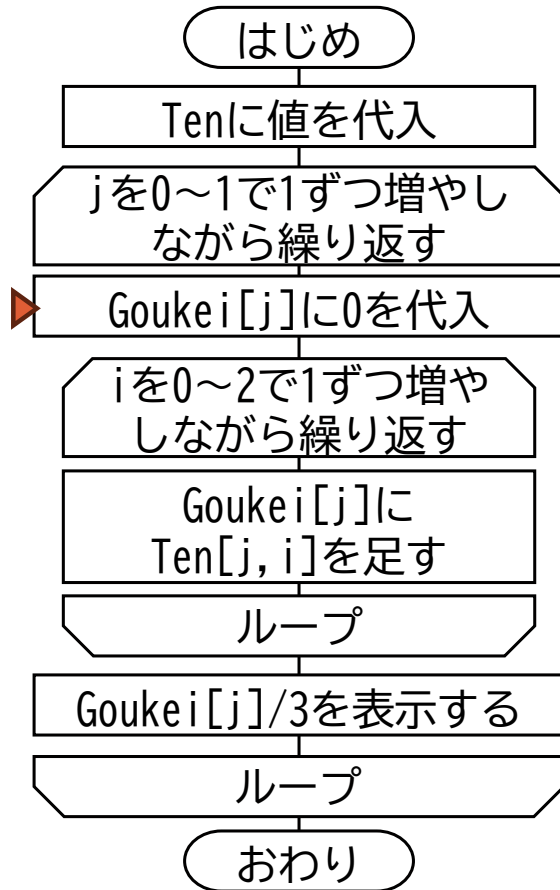
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

	0	1	2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

いまのjは0



変数の意味の把握



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

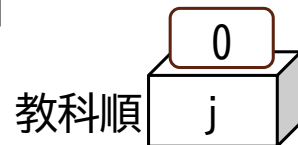
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

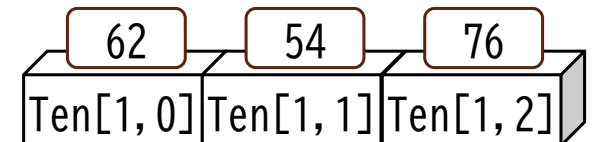
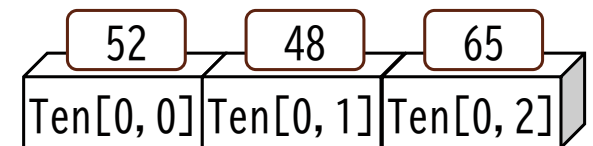
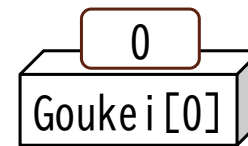
	0	1	2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



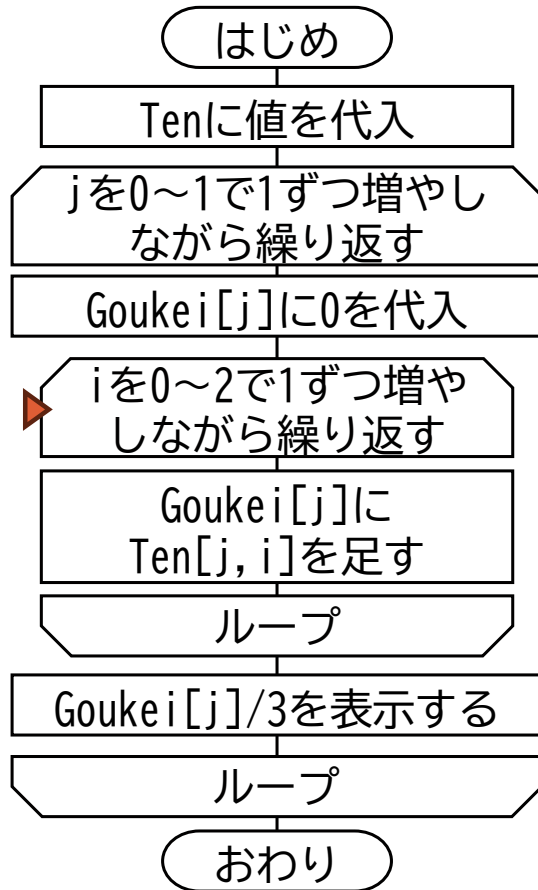
赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



変数の意味の把握



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す :

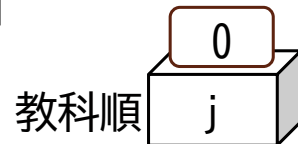
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す :

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	0	1	2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

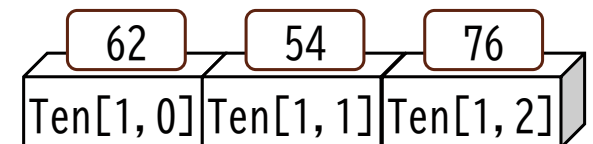
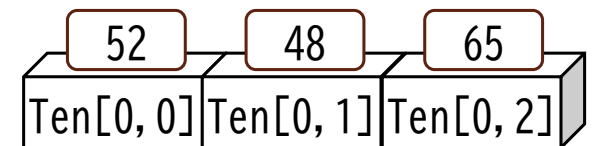
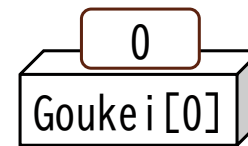


赤字 : 用語も意味も理解が必要、緑字 : 意味や概念の理解が必要

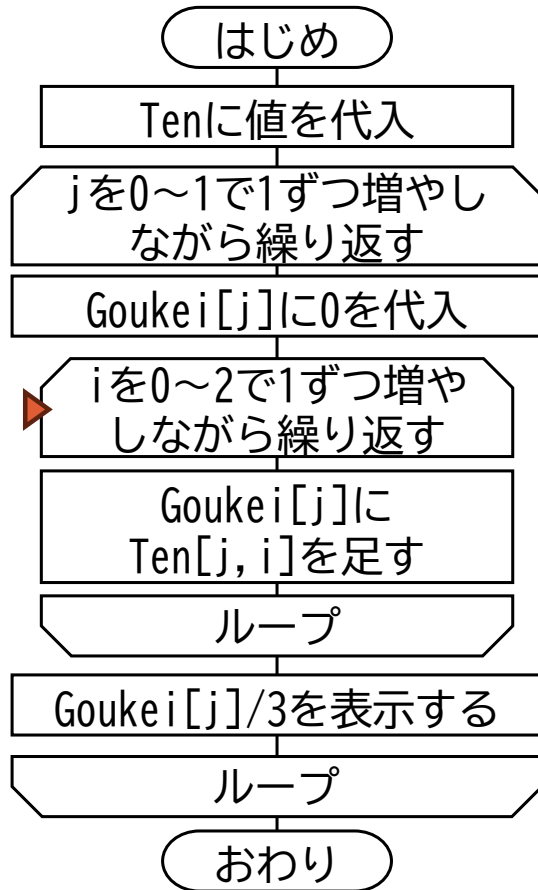
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する

列番号なので、生徒順を示す



変数の意味の把握



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

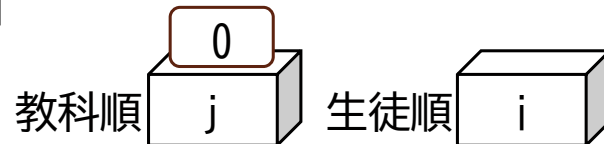
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

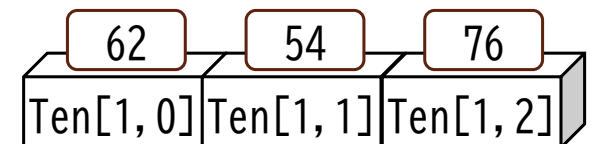
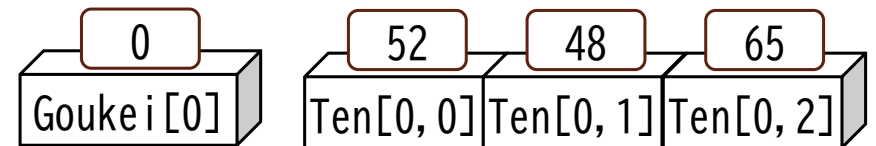


赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

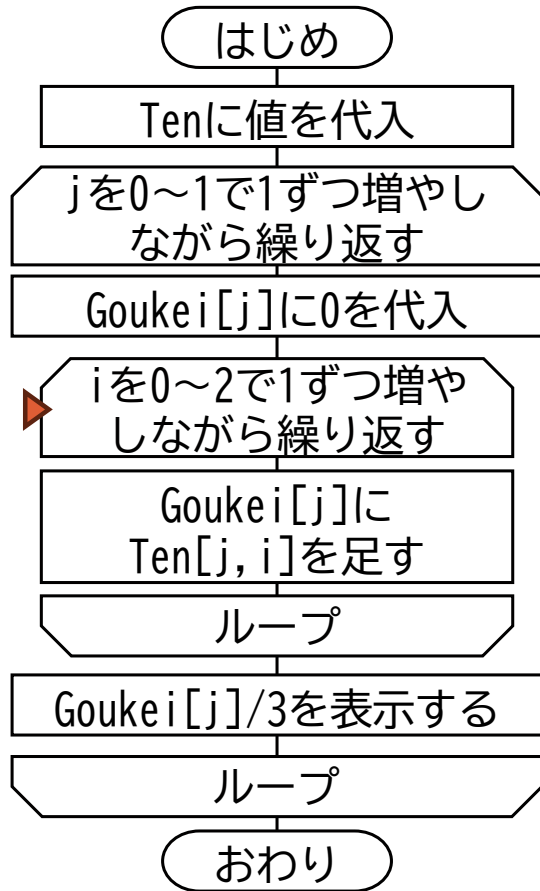
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する

列番号なので、生徒順を示す



変数の意味の把握



共通テストで 사용되는DNCL

```

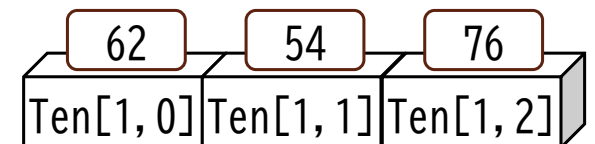
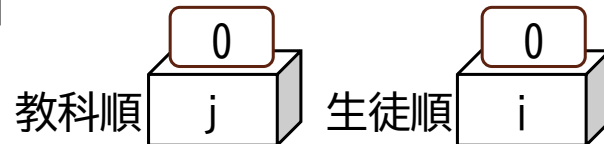
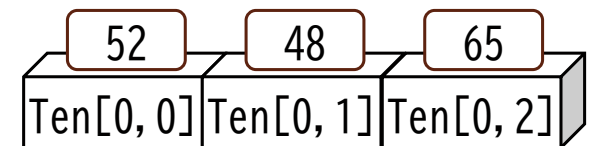
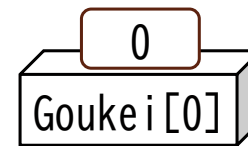
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す :
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す :
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

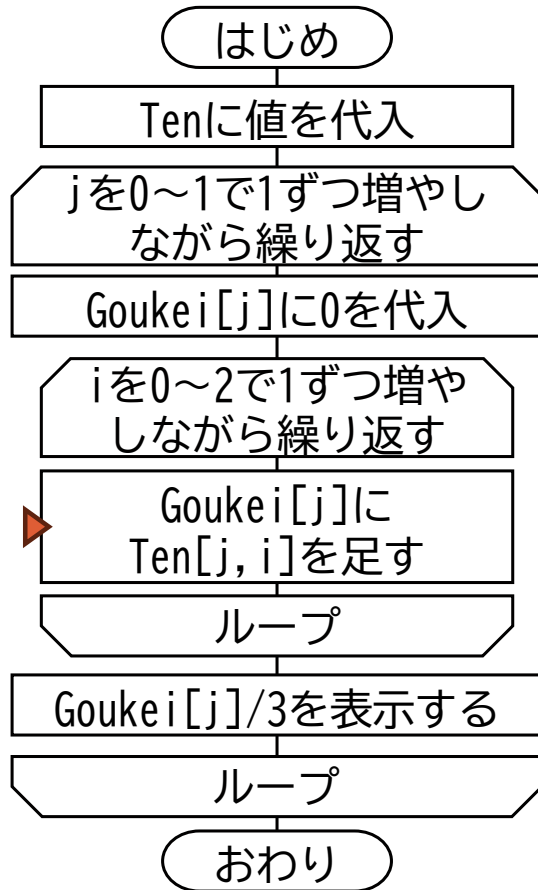
赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

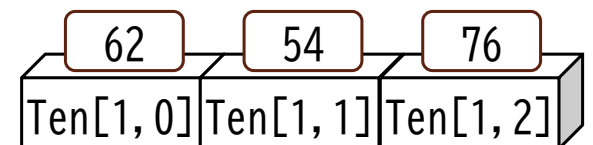
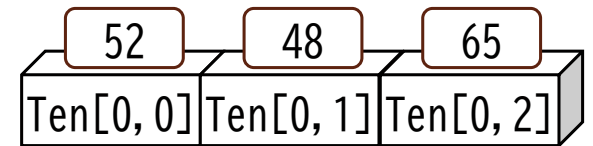
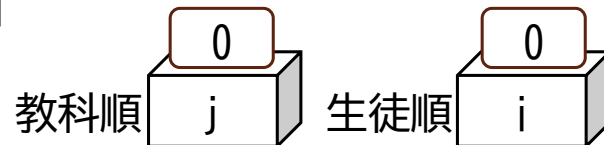
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2	Goukei[j] + Ten[j, i]
生徒	A	B	C	
j=0 英語	52	48	65	0
j=1 数学	62	54	76	

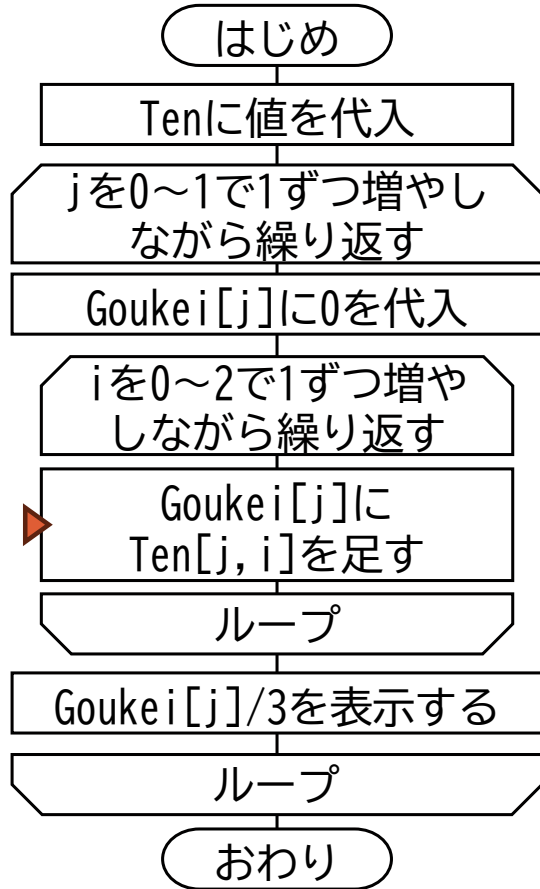


赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する

変数の意味の把握



共通テストで使用されるDNCL

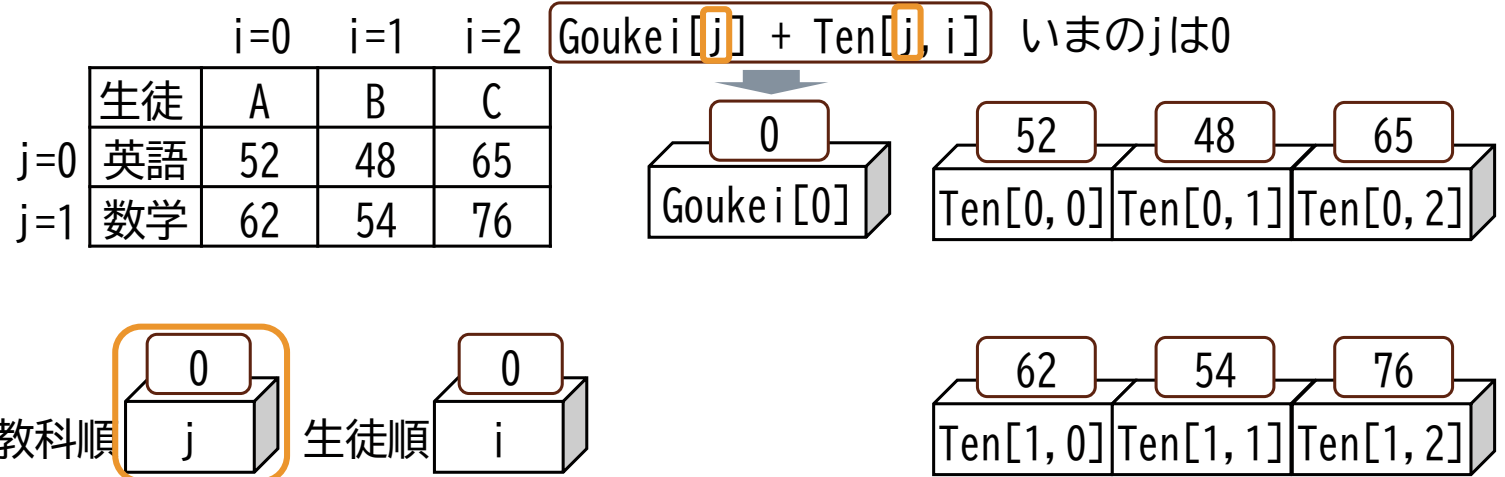
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す :
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す :
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

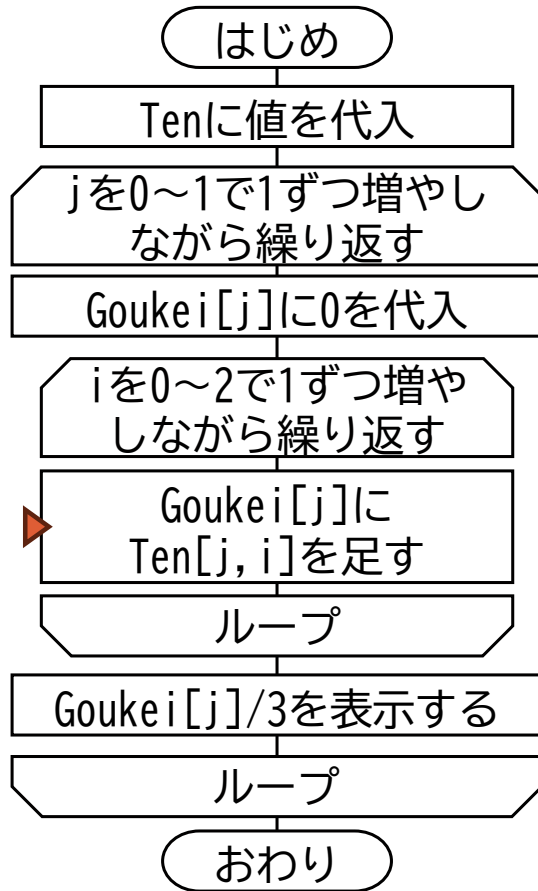
※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



変数の意味の把握



共通テストで使用されるDNCL

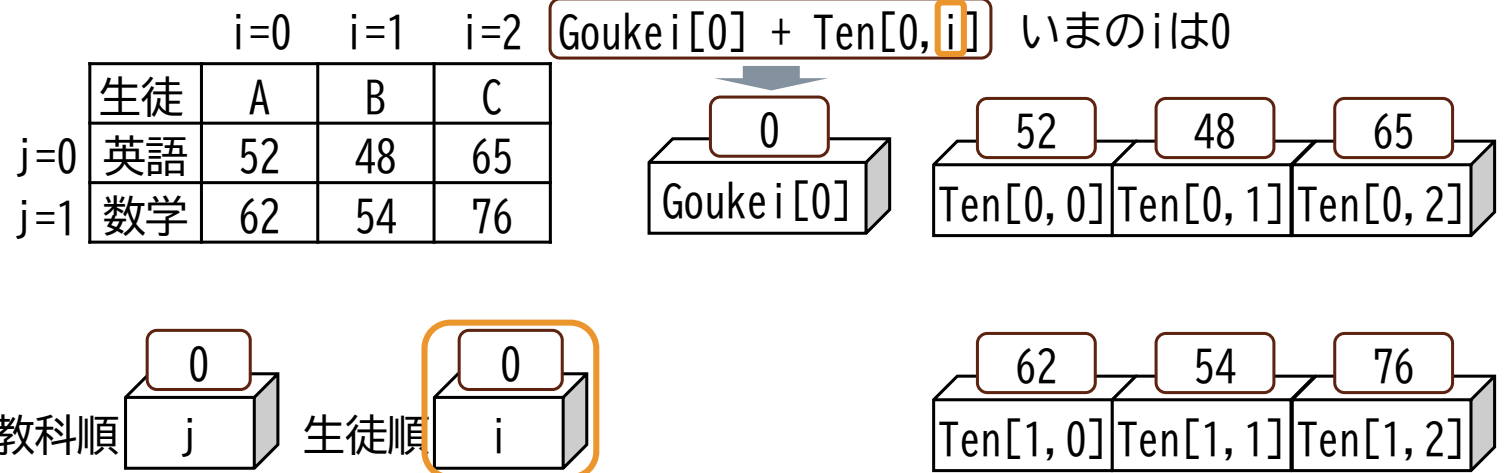
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└ ┌ Goukei[j] = Goukei[j] + Ten[j, i]
  └ 表示する(Goukei[j]/3)
  
```

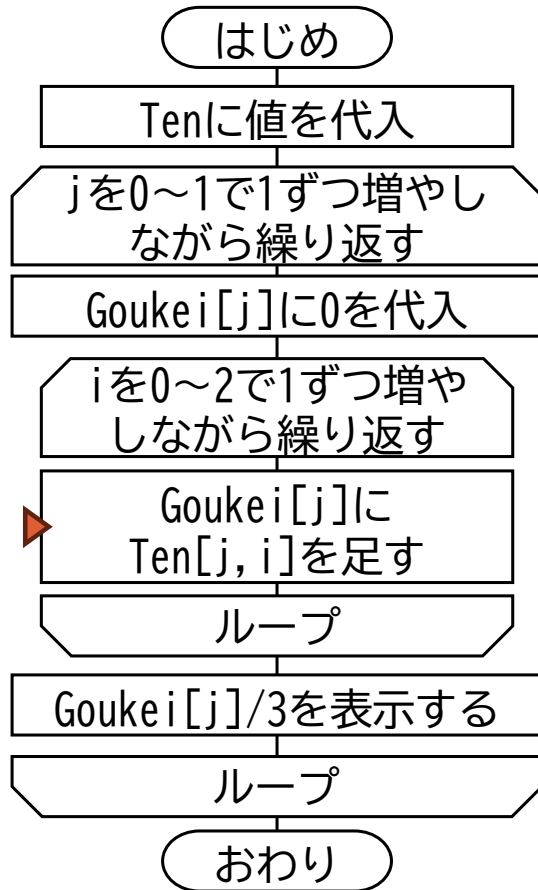
※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



変数の意味の把握



共通テストで 사용되는DNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

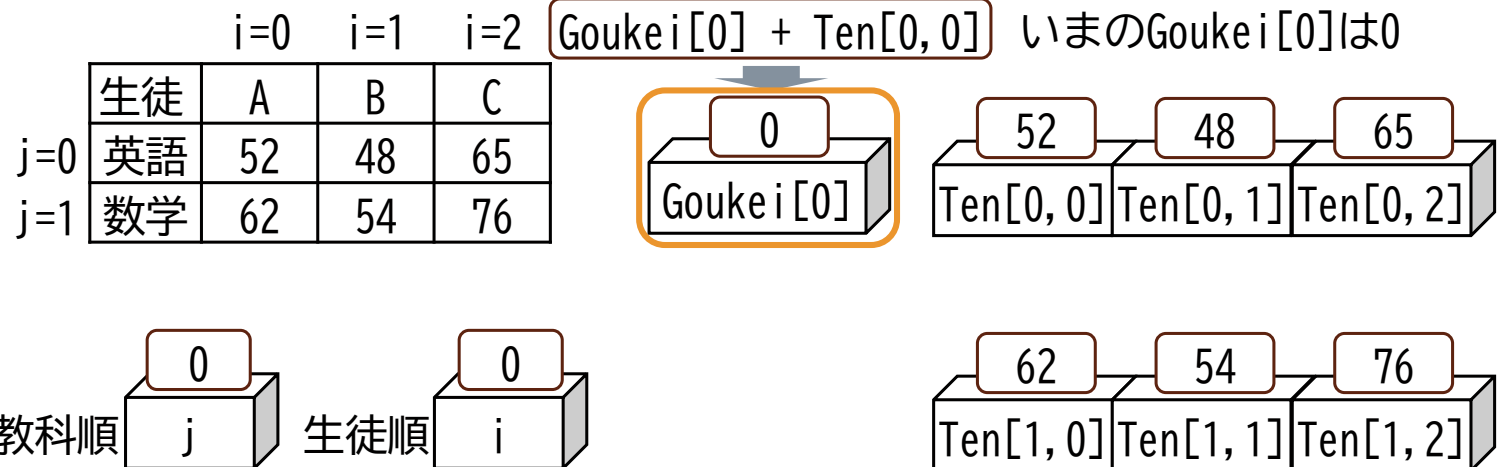
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

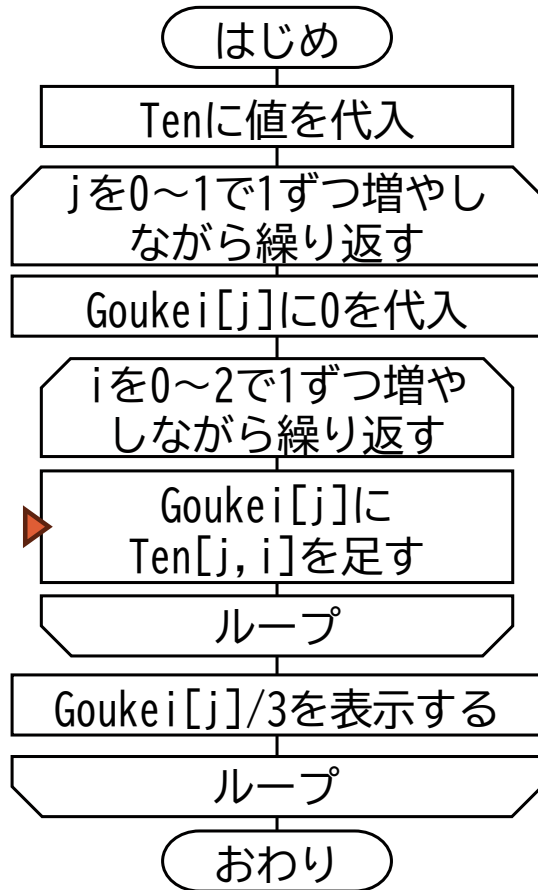
表示する(Goukei[j]/3)

※変数の意味を把握する

- 赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える



変数の意味の把握



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

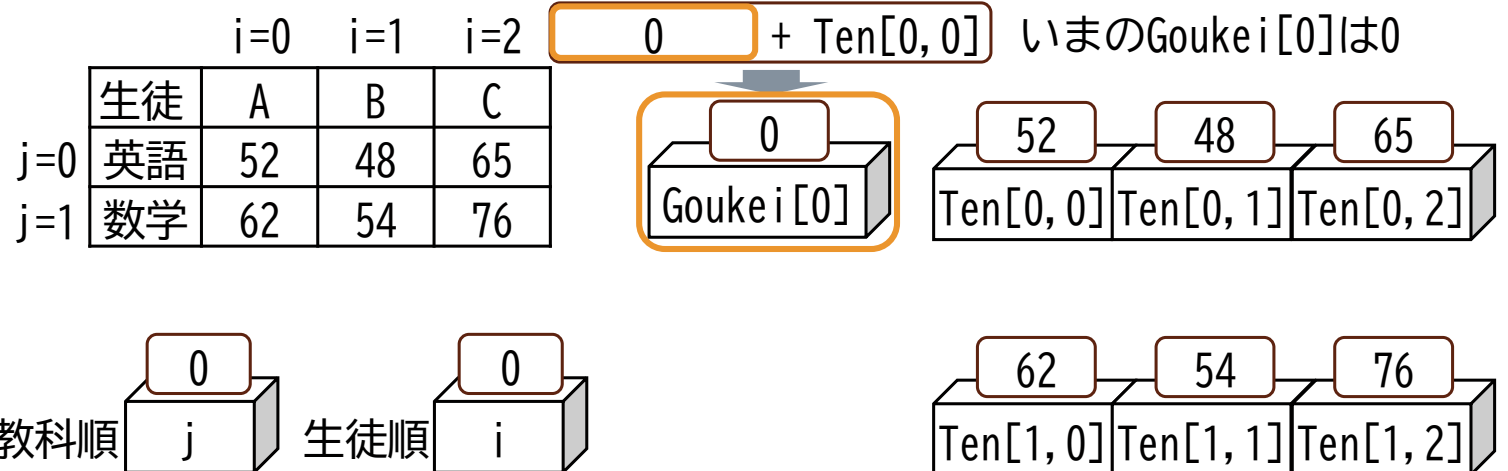
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

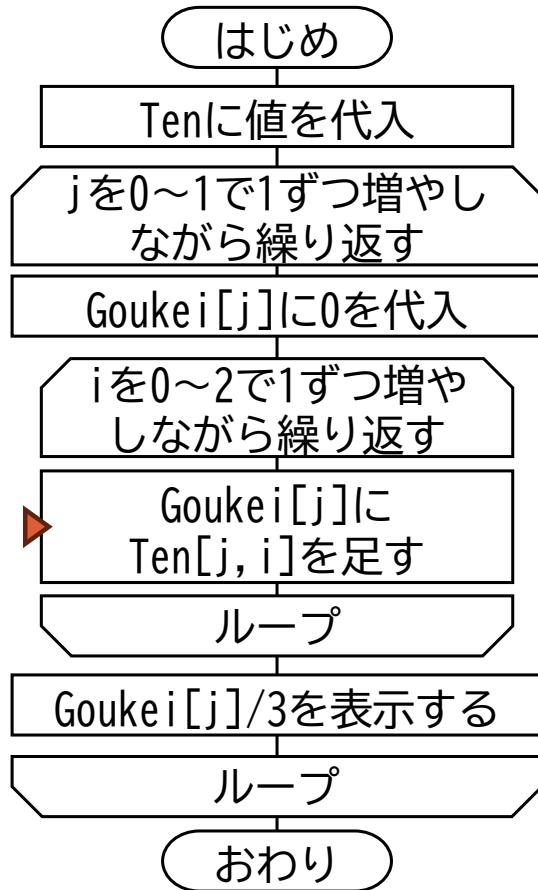
表示する(Goukei[j]/3)

※変数の意味を把握する

- 赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える



変数の意味の把握



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

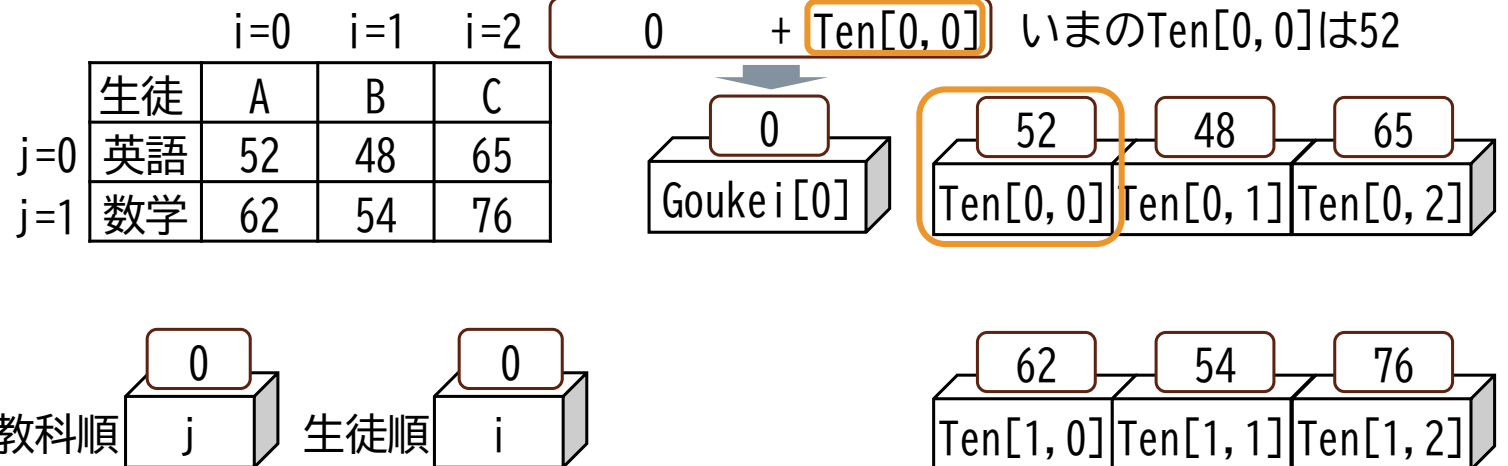
jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

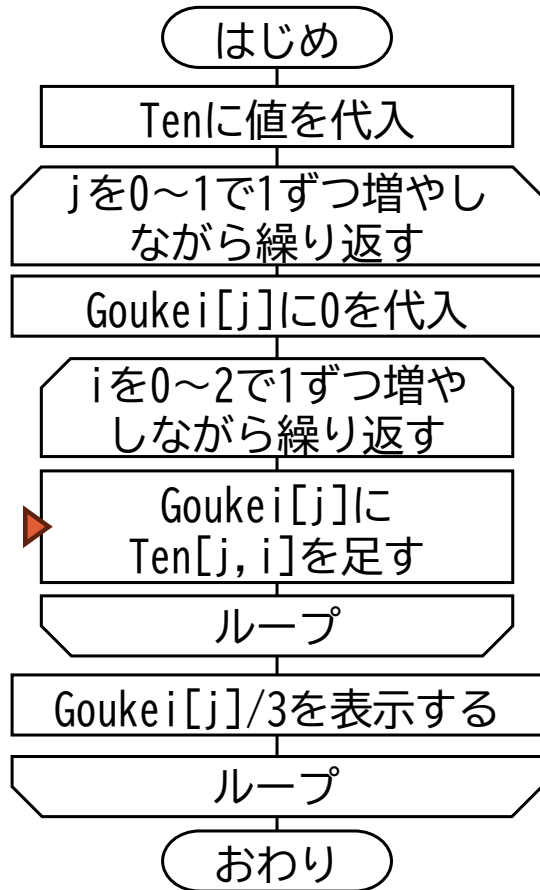


赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える

※変数の意味を把握する

変数の意味の把握



共通テストで使われるDNCL

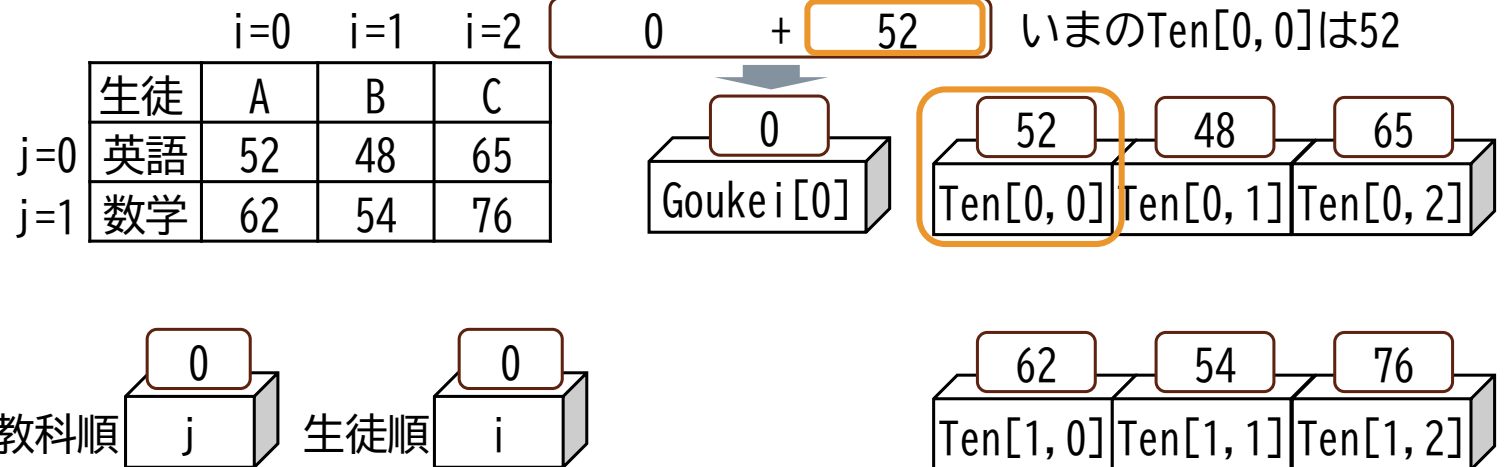
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└ ┌ Goukei[j] = Goukei[j] + Ten[j, i]
  └ 表示する(Goukei[j]/3)
  
```

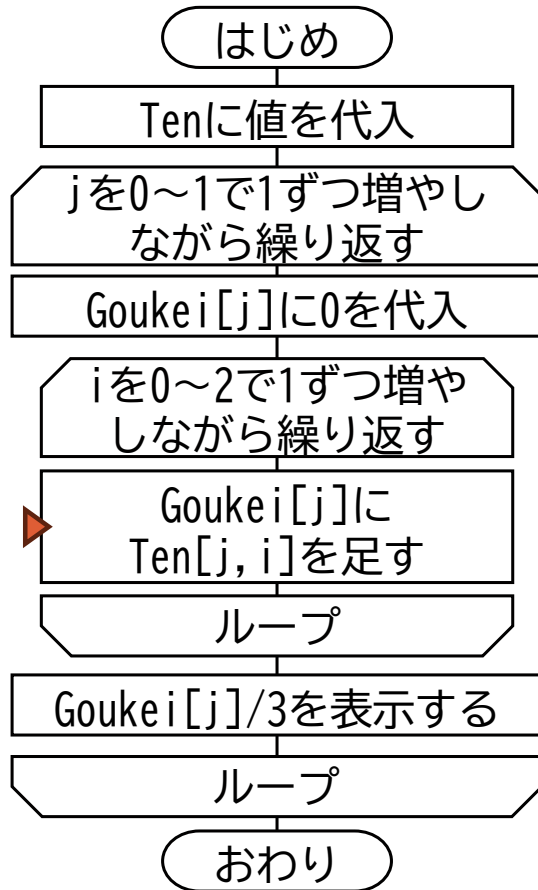
※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える



変数の意味の把握



共通テストで使用されるDNCL

```

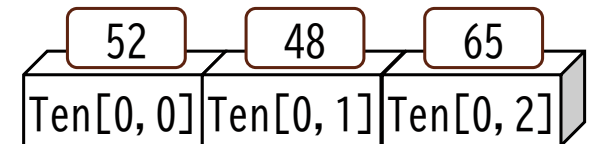
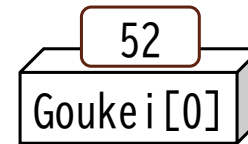
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す :
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す :
└ ┌ Goukei[j] = Goukei[j] + Ten[j, i]
  └ 表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

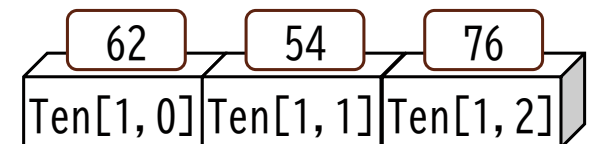
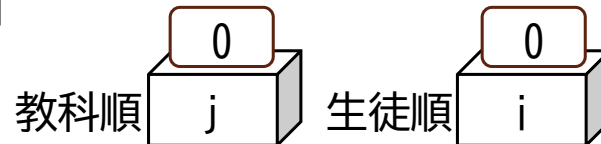
赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



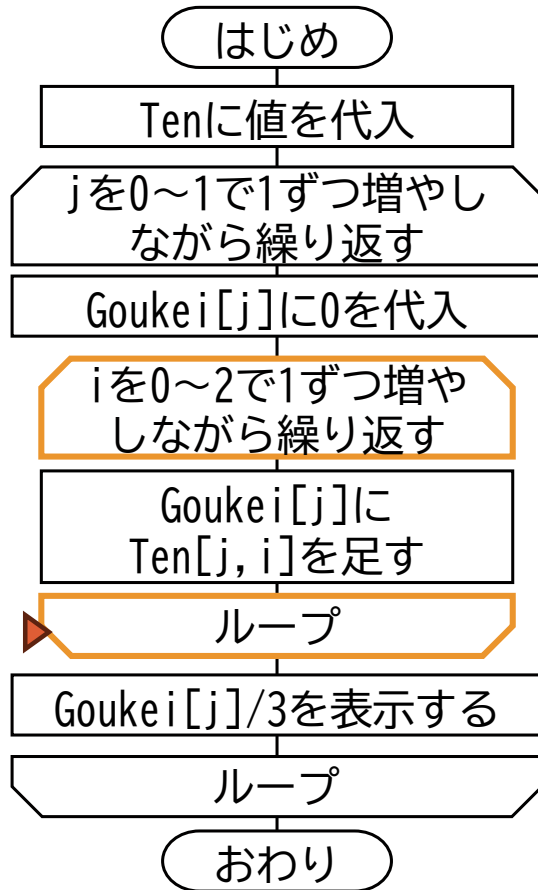
英語の0人目までの合計を計算した



変数の意味の把握、入れ子の反復処理がキモ

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- ・ 「=」は代入の意味 ※等しいではない
- ・ 計算をしてから変数に代入する
 - ・ 変数は値に置き換える



共通テストで使用されるDNCL

```

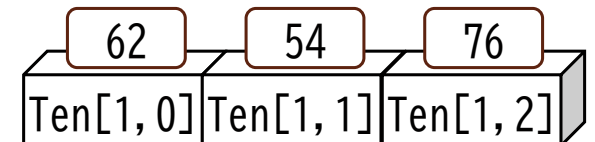
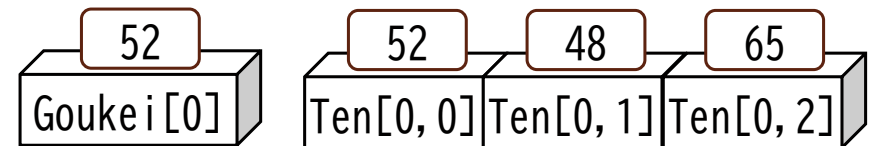
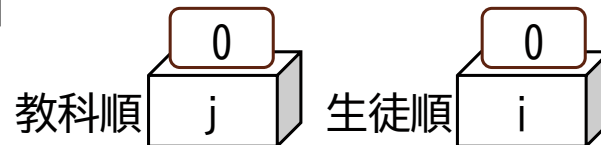
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)

```

※変数の意味を把握する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

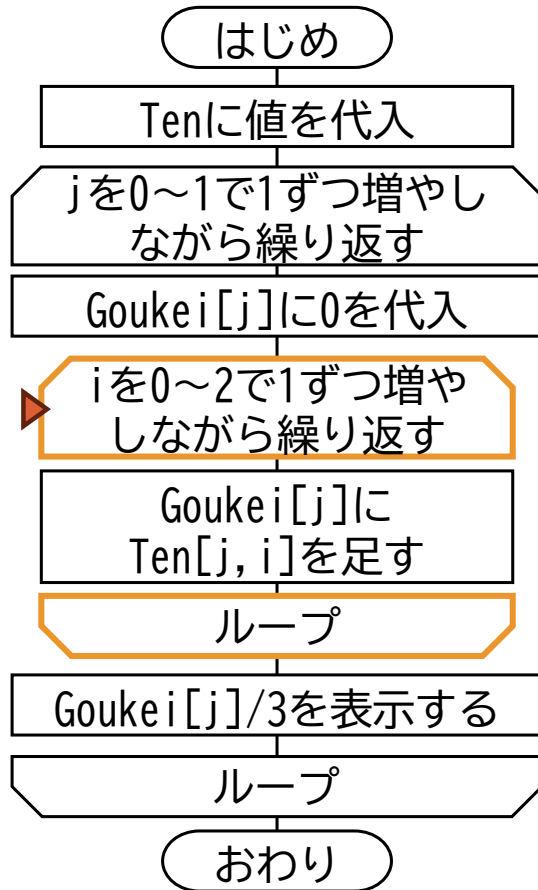
反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで 사용되는DNCL

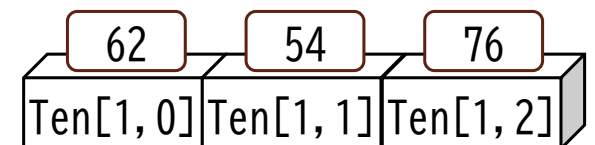
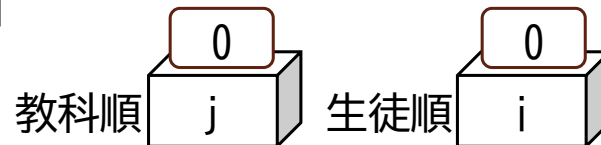
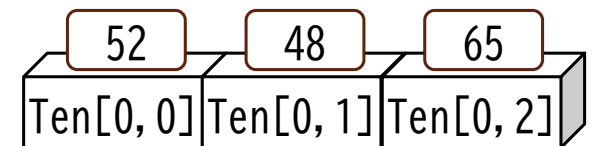
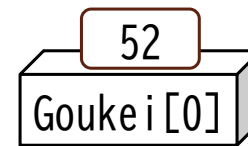
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)

```

反復処理の終了部分にきたら開始部分に戻る

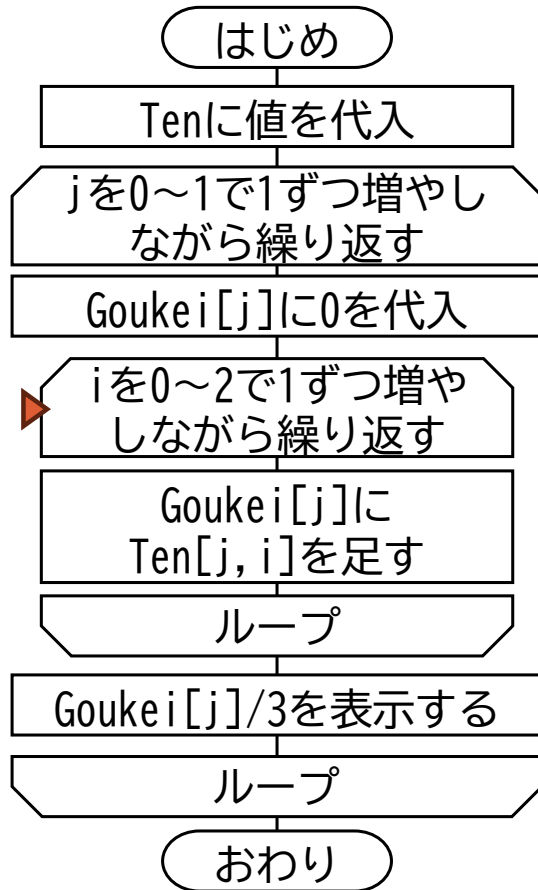
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

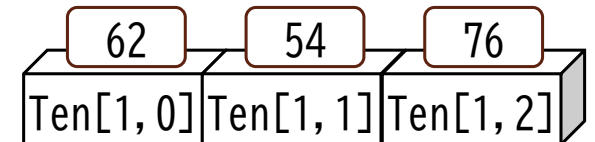
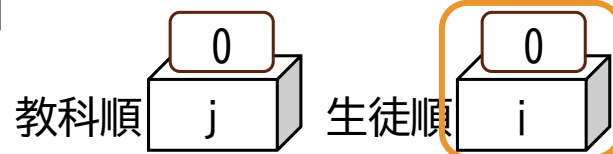
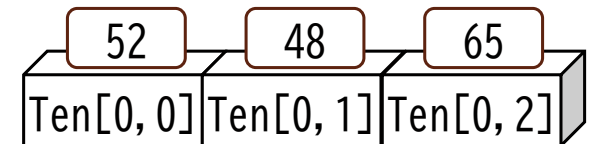
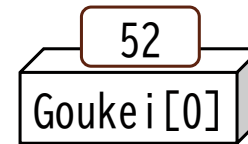
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：………いまのiは0

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

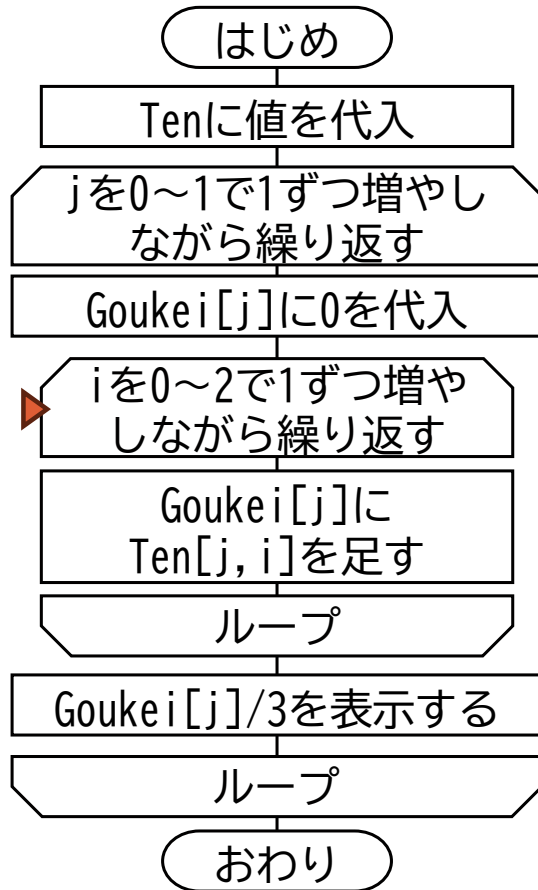
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

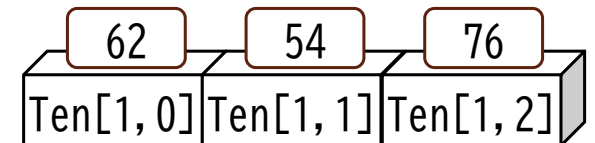
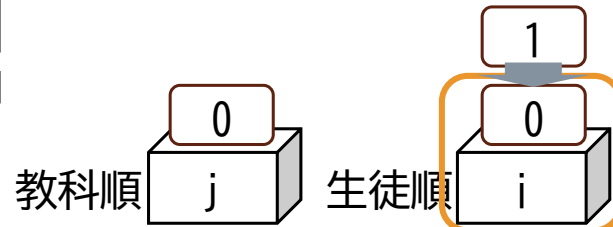
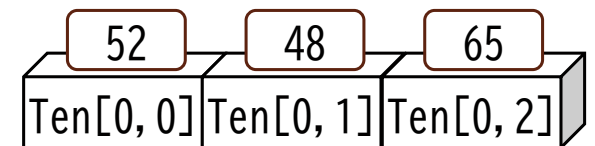
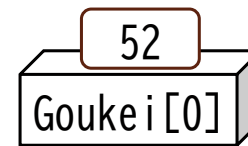
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：………いまのiは0

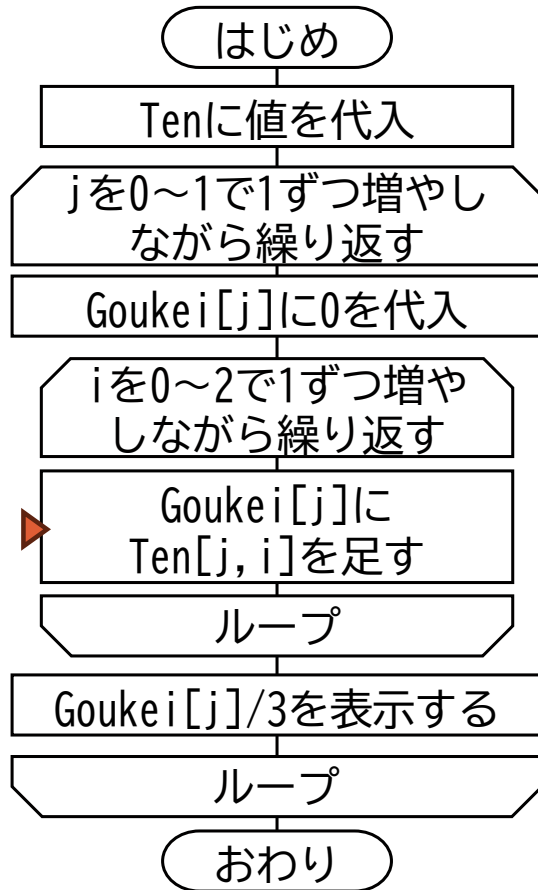
Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

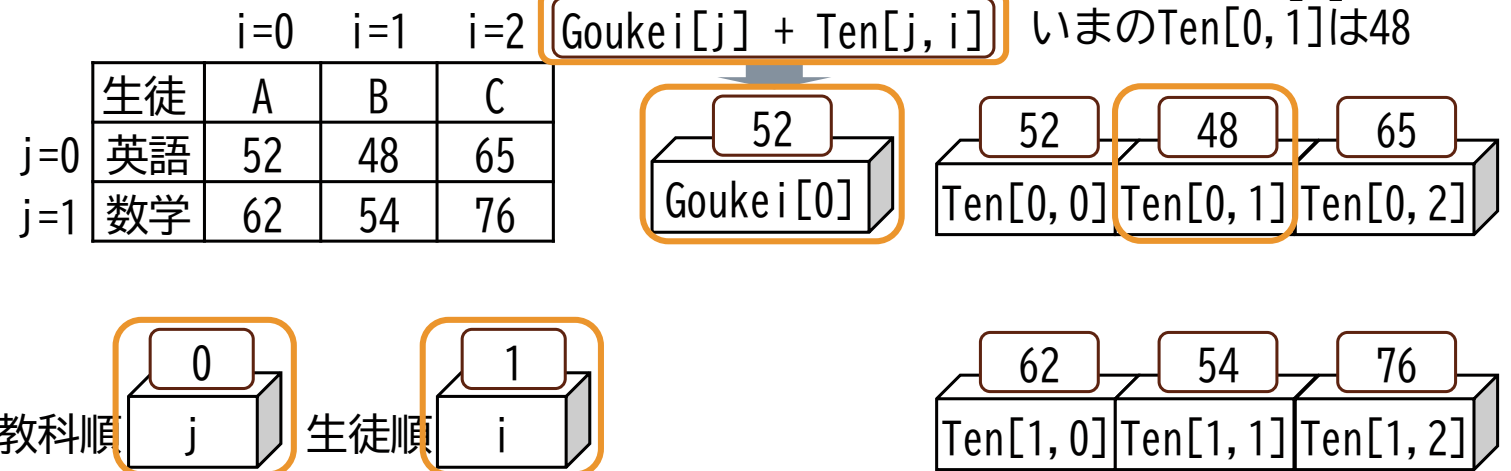
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

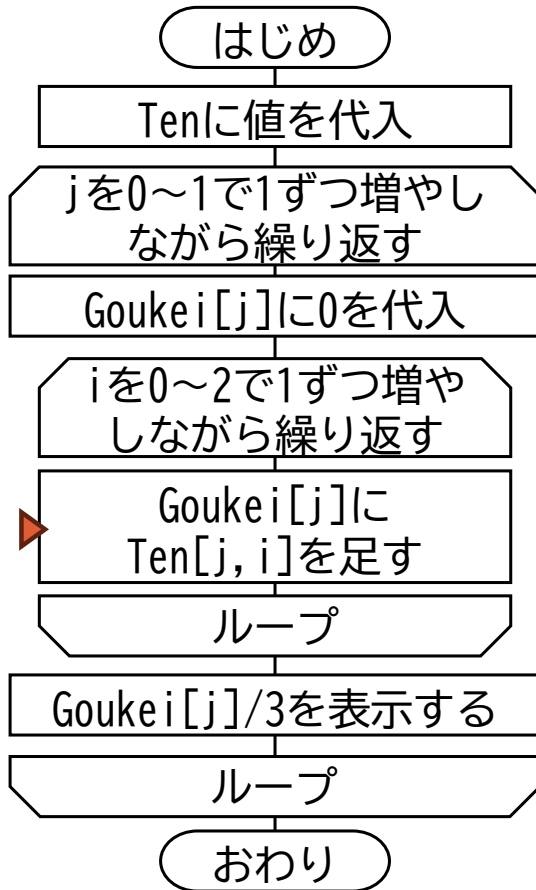
表示する(Goukei[j]/3)

※変数の意味を把握する



いまのjは0、いまのiは1
いまのGoukei[0]は52
いまのTen[0, 1]は48

変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながらか繰り返す：

Goukei[j] = 0

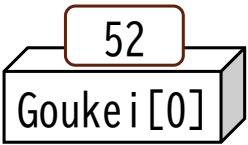
iを0から2まで1ずつ増やしながらか繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

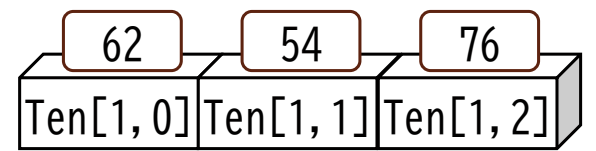
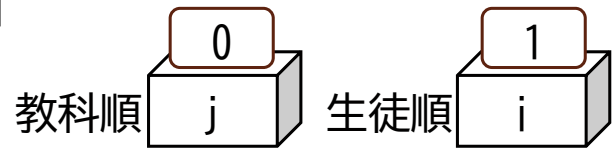
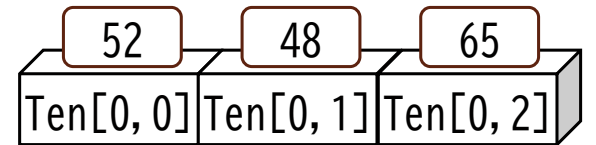
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

52 + 48



※変数の意味を把握する

いまのjは0、いまのiは1
いまのGoukei[0]は52
いまのTen[0, 1]は48

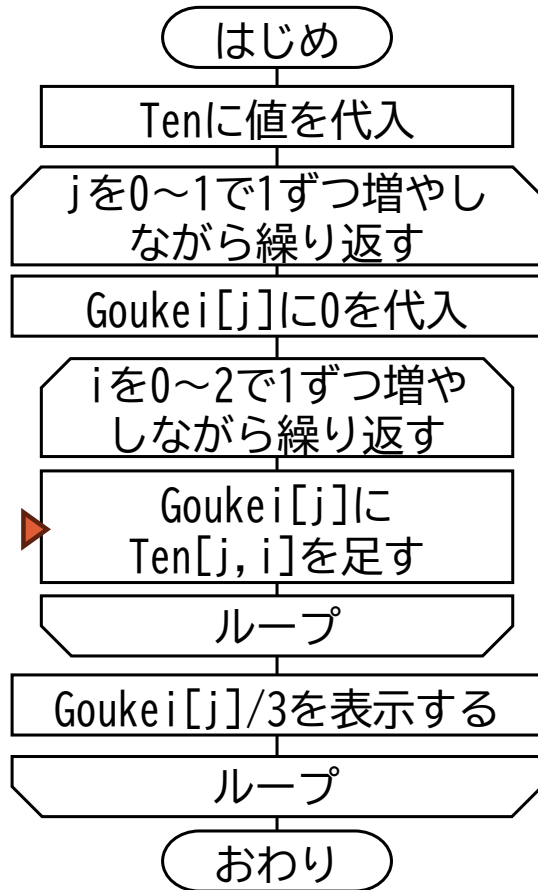


- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える

変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



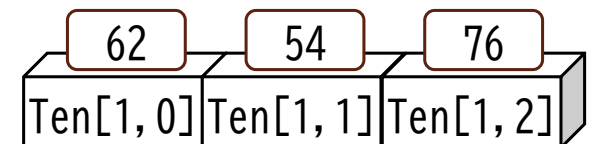
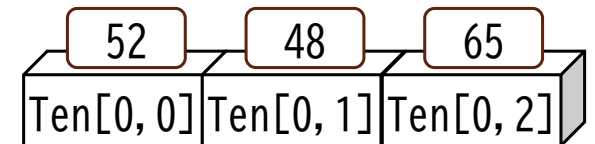
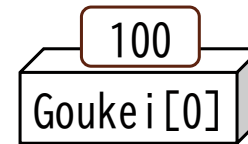
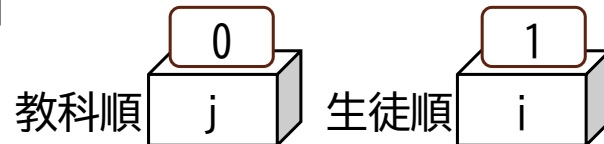
共通テストで使用されるDNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

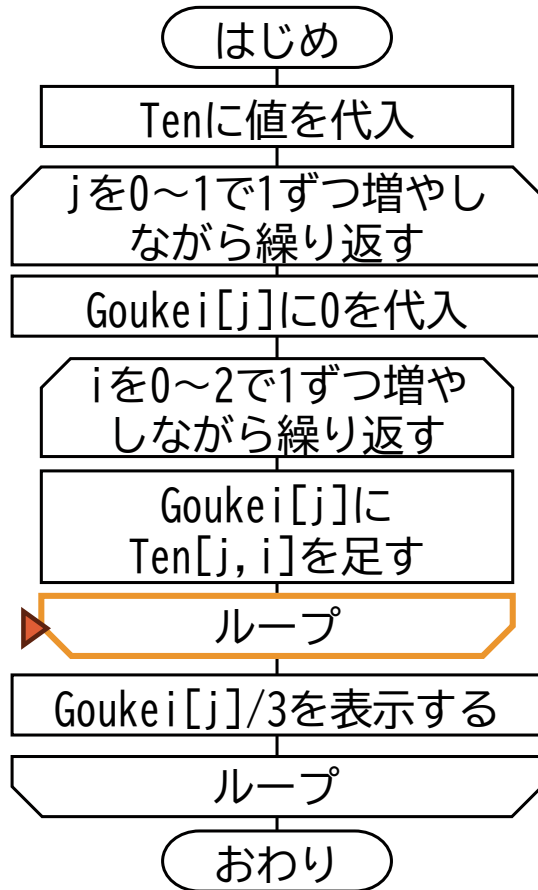
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

英語の0~1人目までの合計を計算した



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで 사용되는DNCL

```

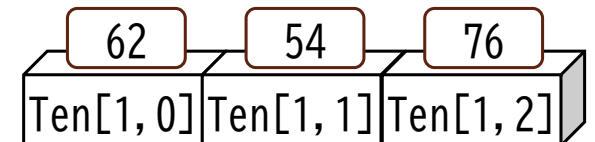
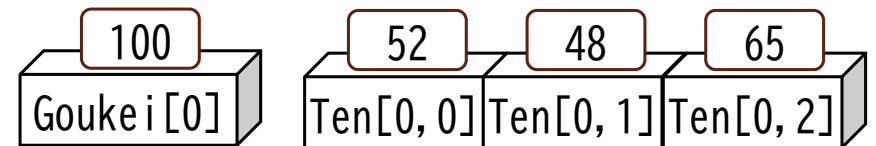
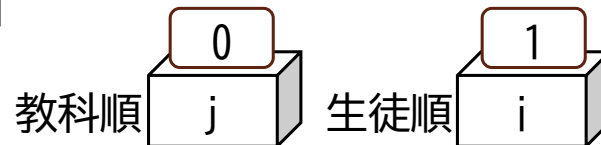
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)

```

※変数の意味を把握する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

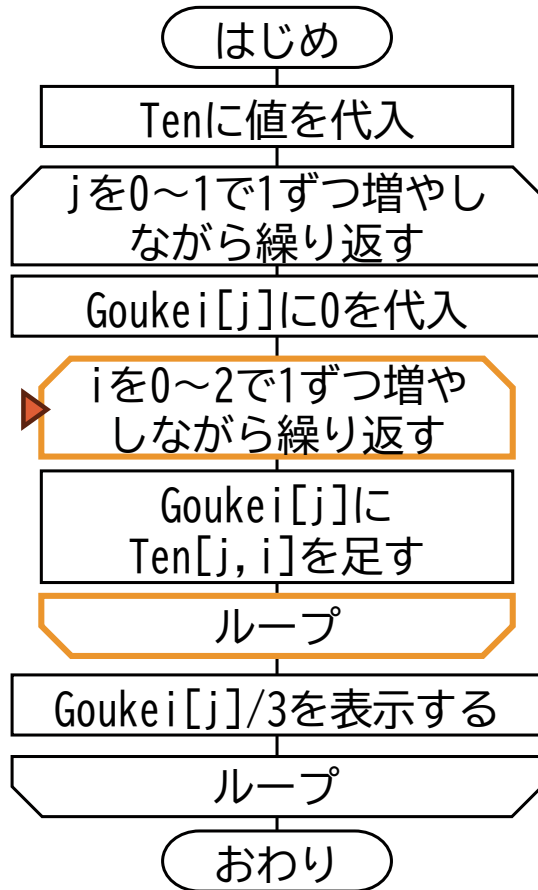
反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで 사용되는DNCL

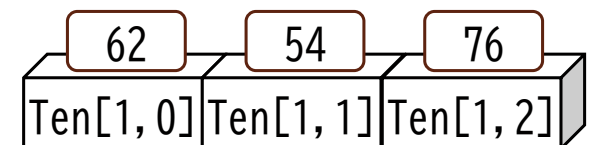
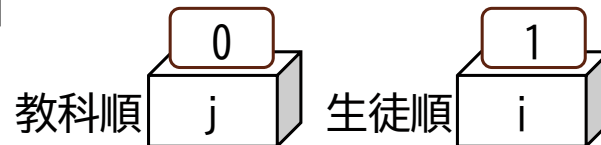
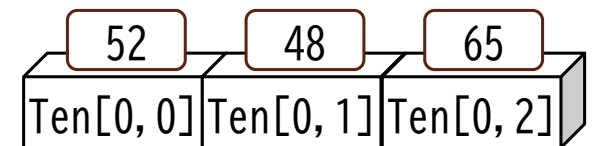
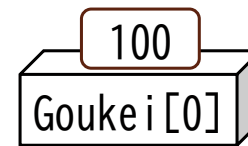
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)

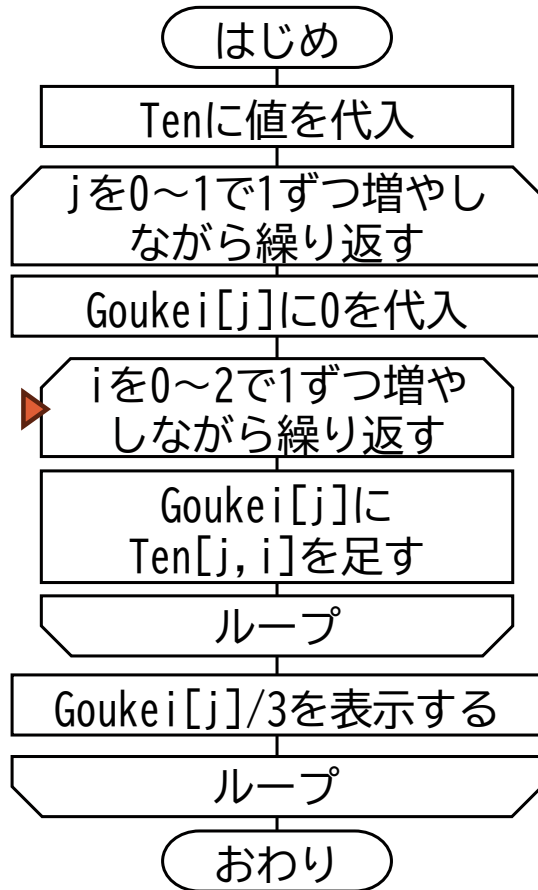
```

反復処理の終了部分にきたら開始部分に戻る

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

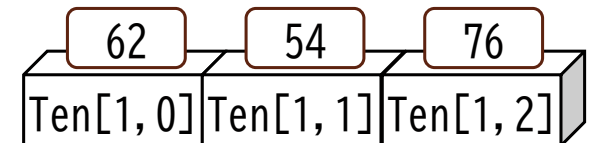
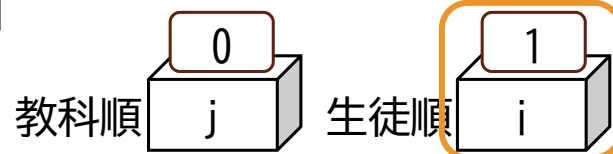
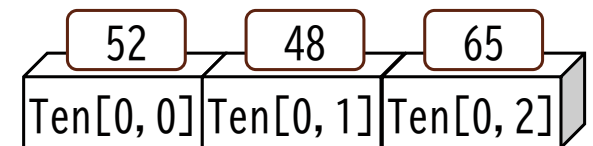
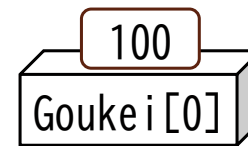
iを0から2まで1ずつ増やしながら繰り返す：………いまのiは1

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

※変数の意味を把握する

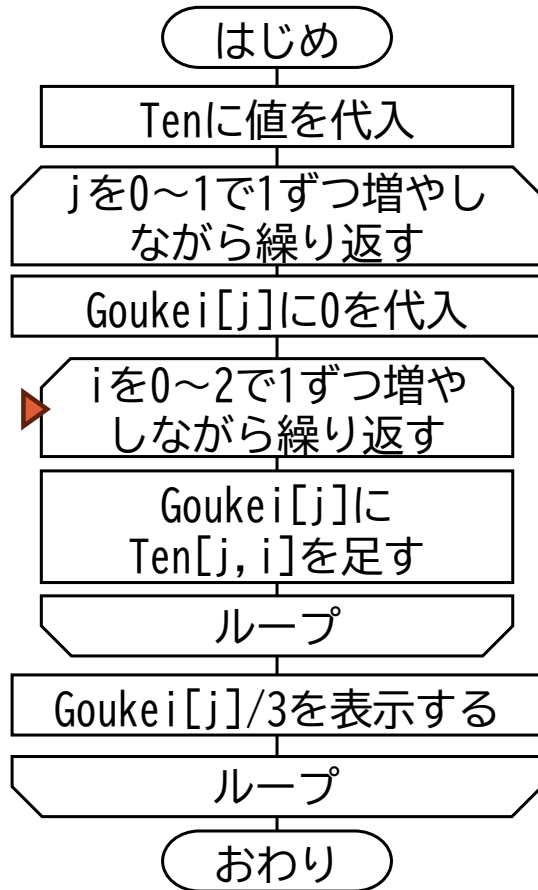
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

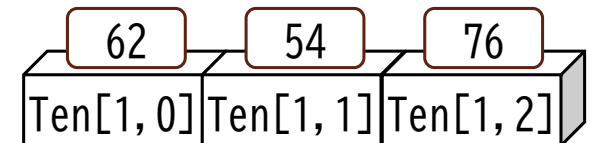
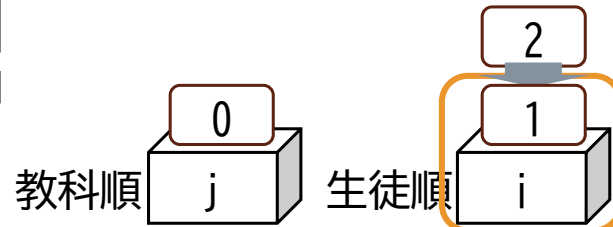
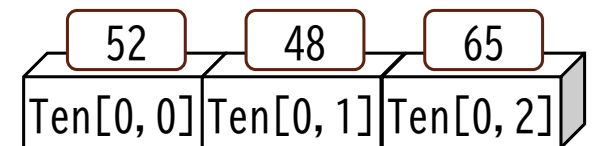
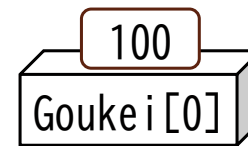
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：………いまのiは1

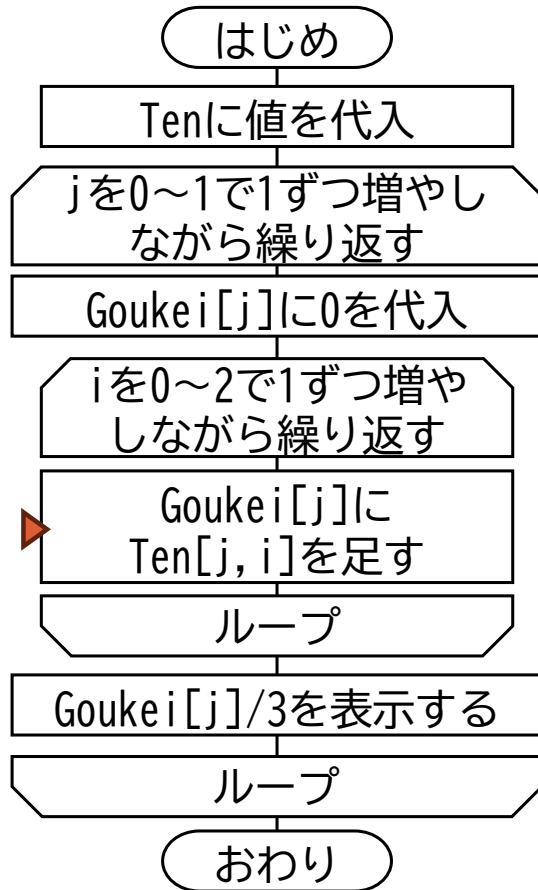
Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

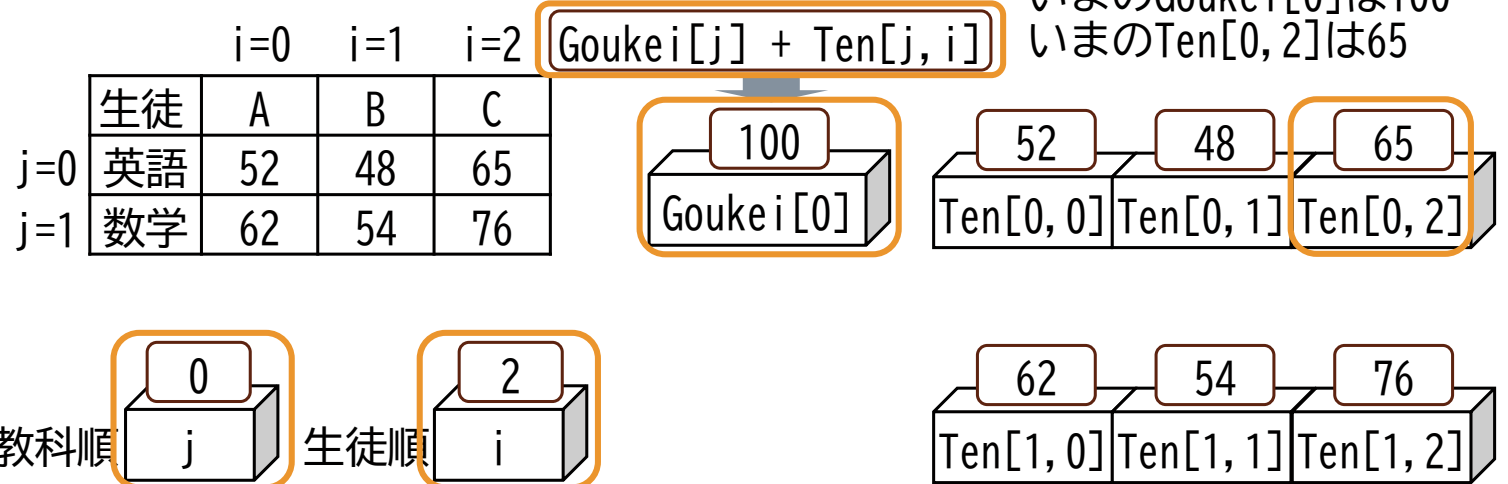
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

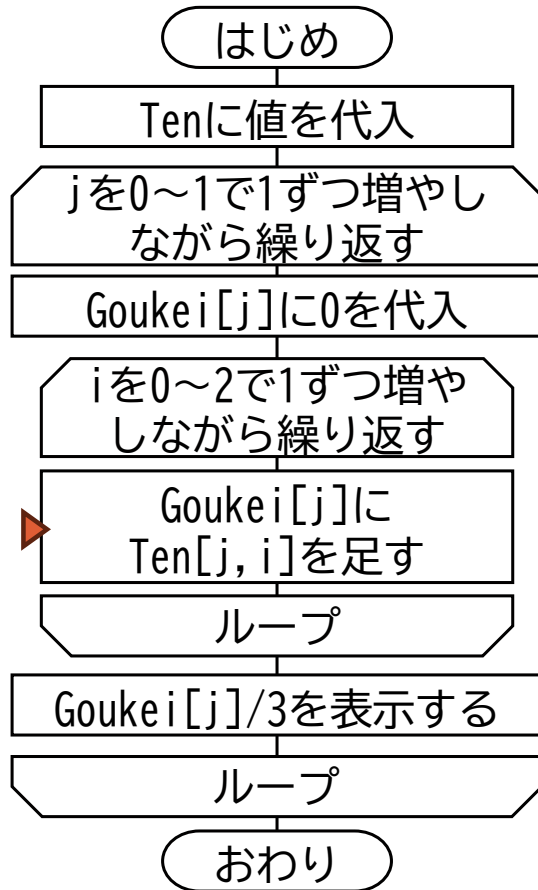
Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

※変数の意味を把握する



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

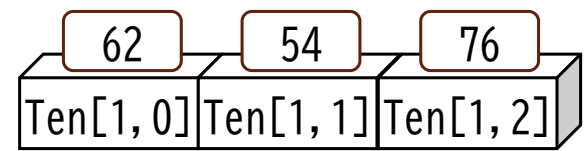
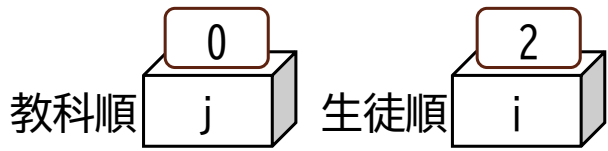
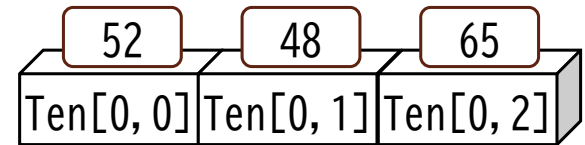
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

100 + 65



※変数の意味を把握する

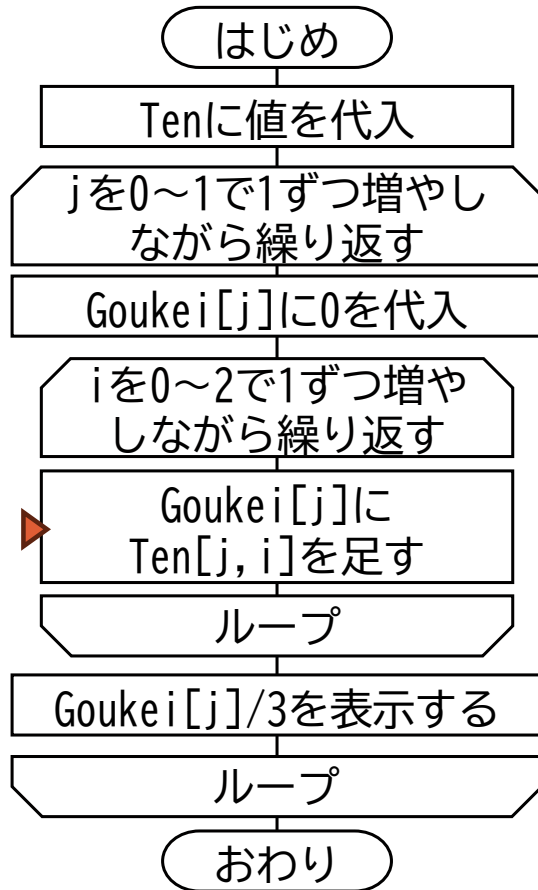
いまのjは0、いまのiは2
いまのGoukei[0]は100
いまのTen[0, 2]は65

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する

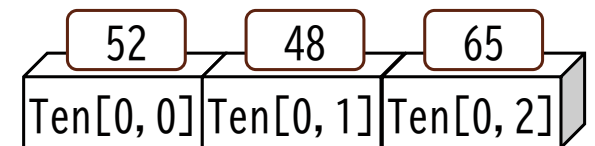
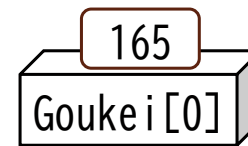


共通テストで使用されるDNCL

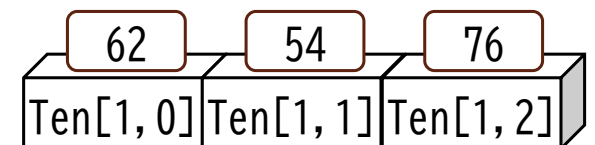
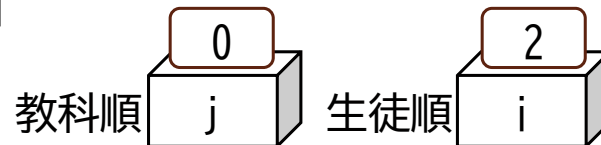
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながらか繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながらか繰り返す：
└ Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

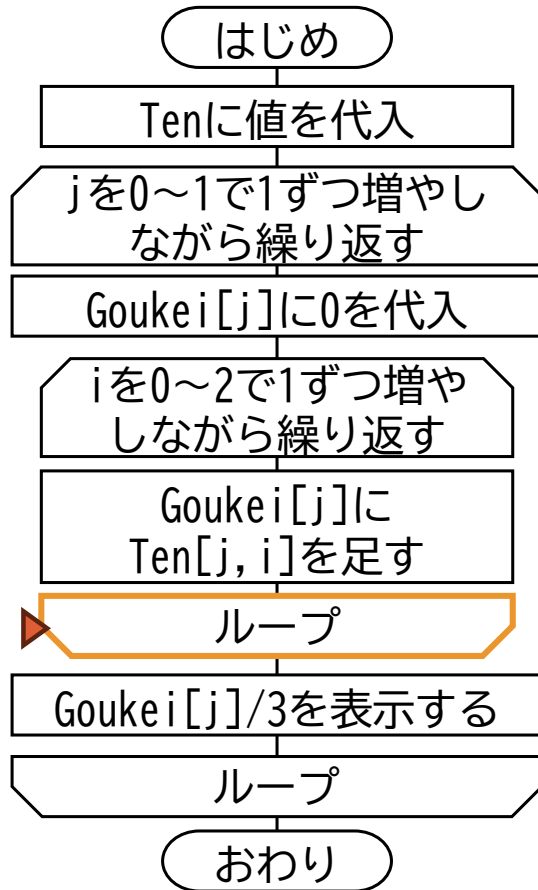
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



英語の0~2人目までの合計を計算した



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

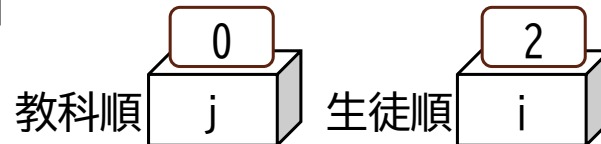
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

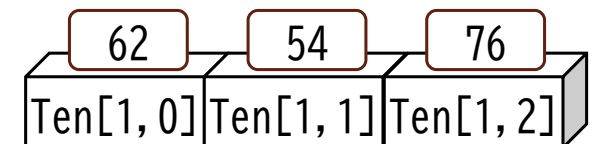
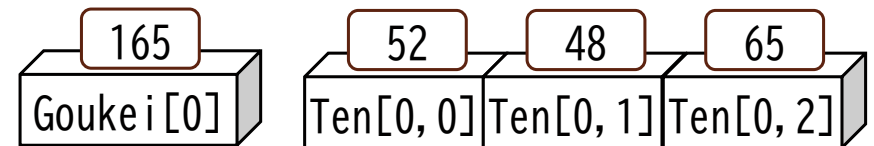
反復処理の終了部分にきたら開始部分に戻る



赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える

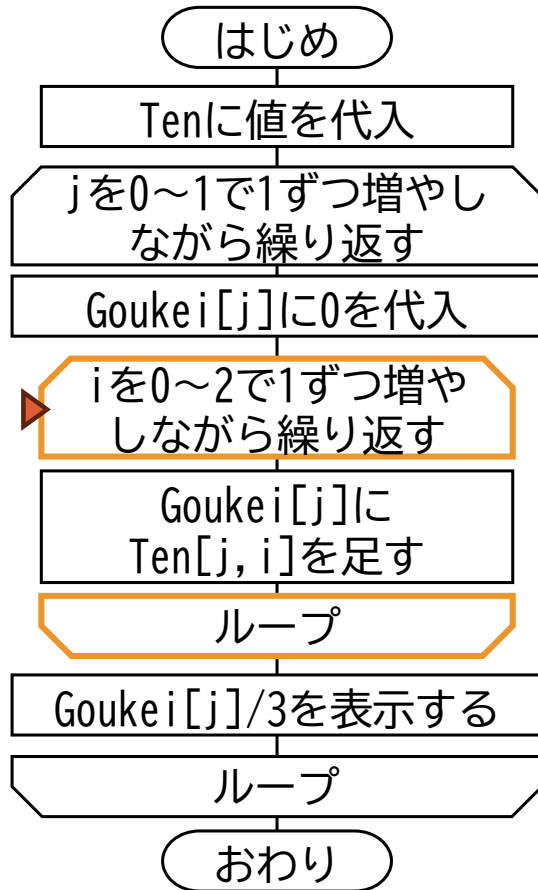
※変数の意味を把握する



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで使われるDNCL

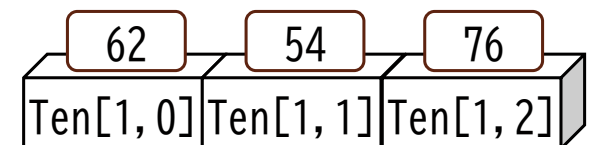
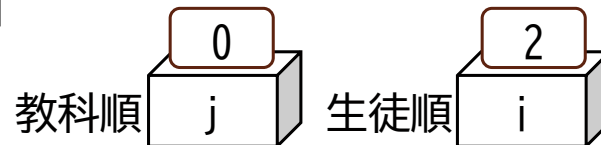
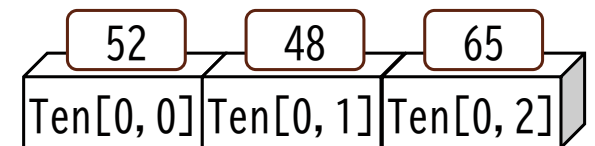
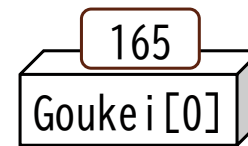
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

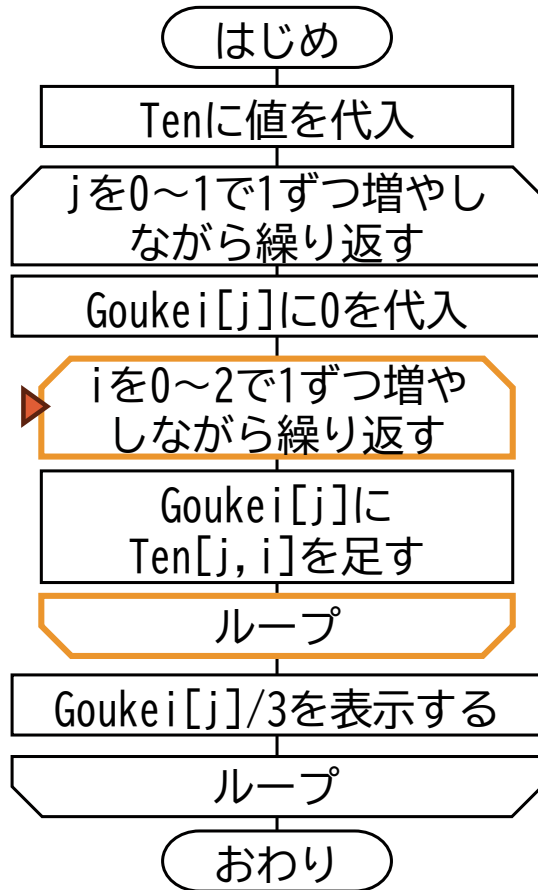
反復処理の終了部分にきたら開始部分に戻る

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで 사용되는DNCL

```

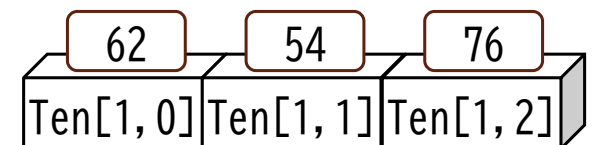
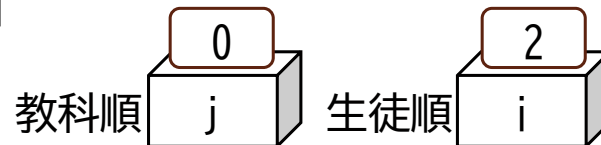
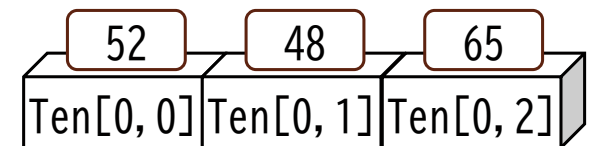
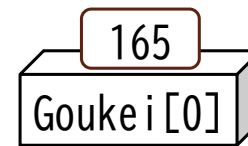
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：………いまのiは2
  Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

※変数の意味を把握する

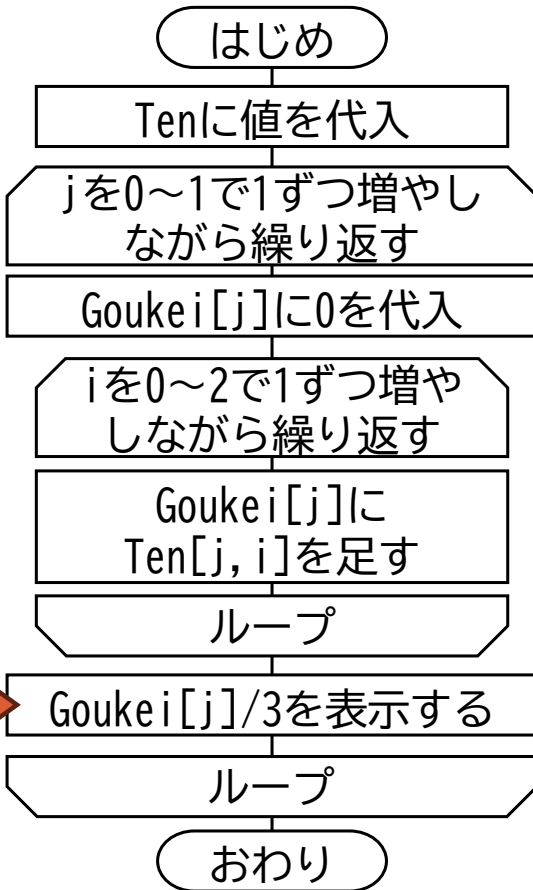
上限に達したので
反復処理の次に進む

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



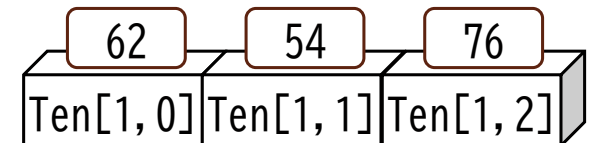
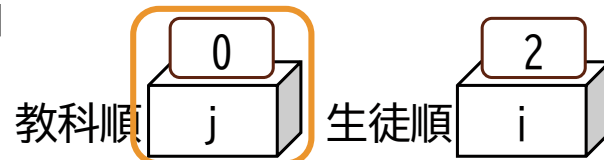
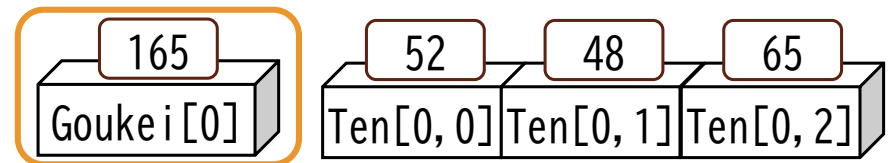
共通テストで使われるDNCL

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
    Goukei[j] = 0
    iを0から2まで1ずつ増やしながら繰り返す：
        Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)
```

※変数の意味を把握する

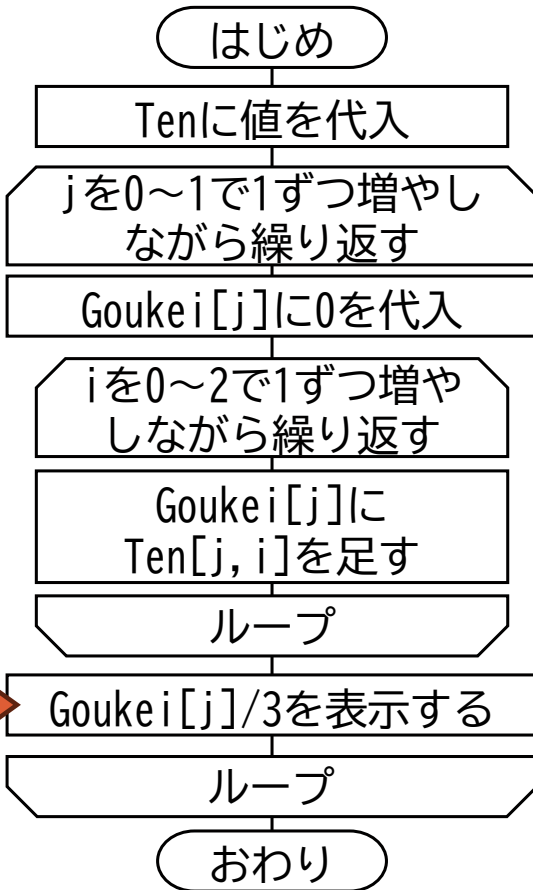
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

いまのjは0
いまのGoukei[0]は165
165/3=55



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

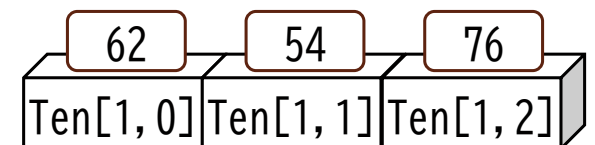
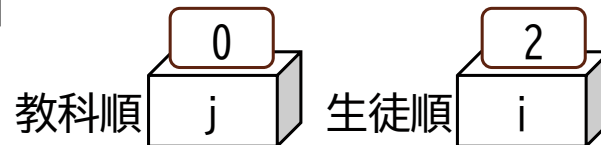
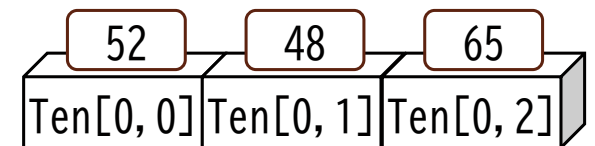
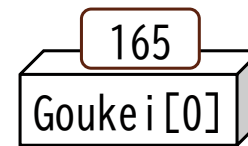
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

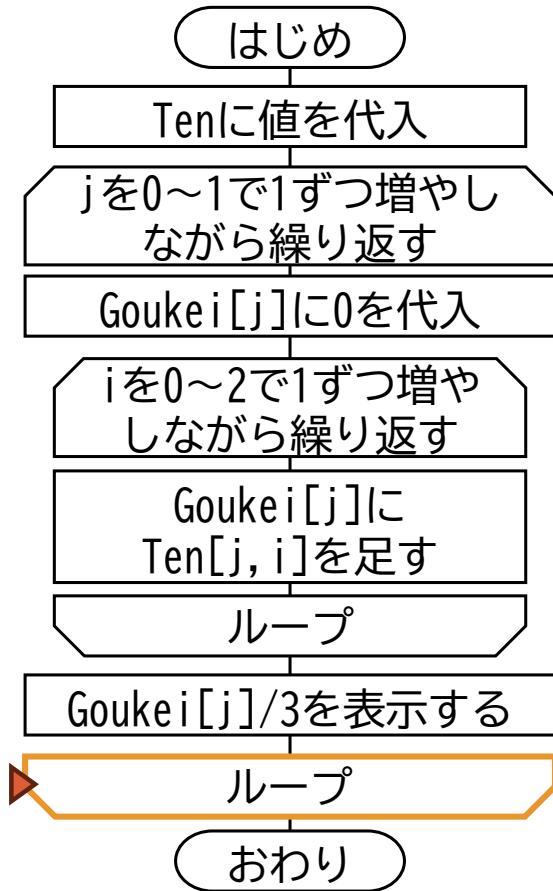
表示する(Goukei[j]/3)..... 「55」と表示される

※変数の意味を把握する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

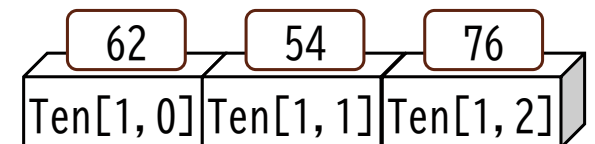
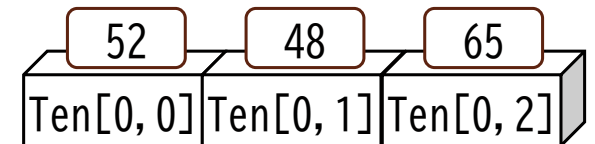
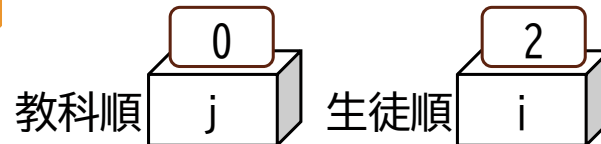
※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

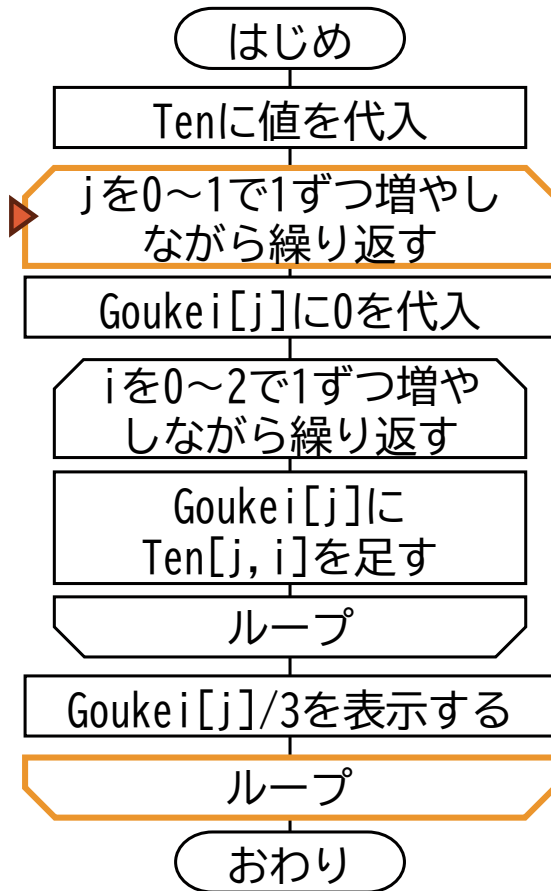
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ



共通テストで 사용되는DNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
▶ jを0から1まで1ずつ増やしながら繰り返す：
   Goukei[j] = 0
   iを0から2まで1ずつ増やしながら繰り返す：
   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
   └ 表示する(Goukei[j]/3)

```

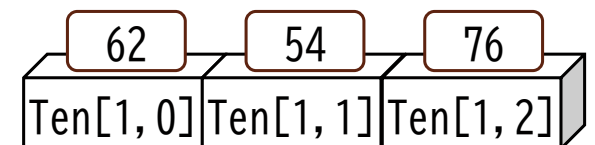
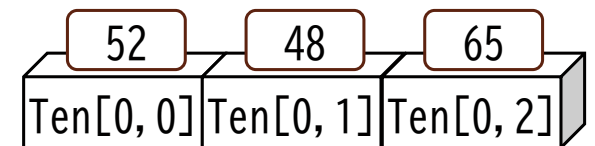
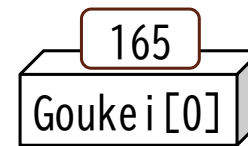
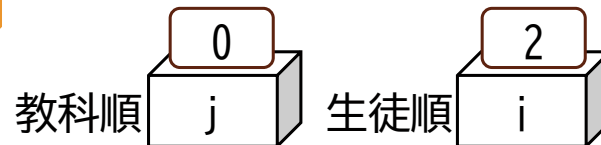
※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

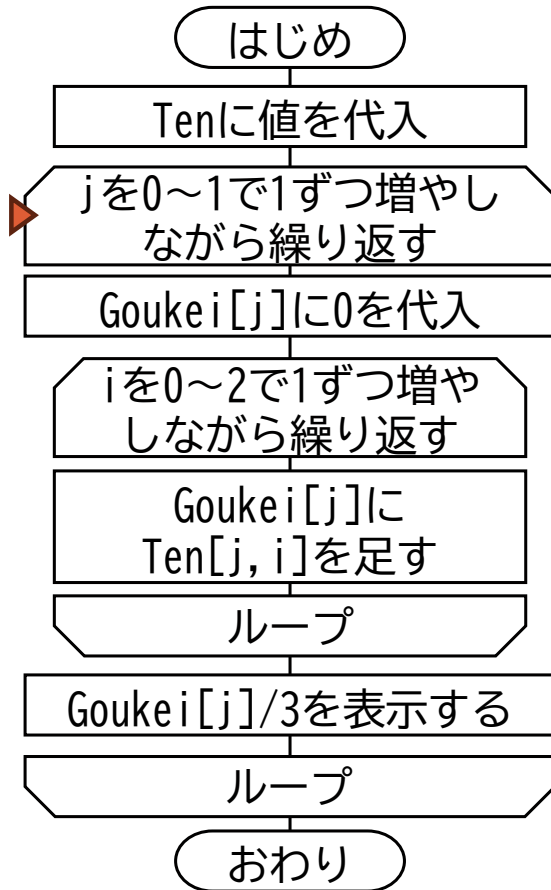
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使われるDNCL

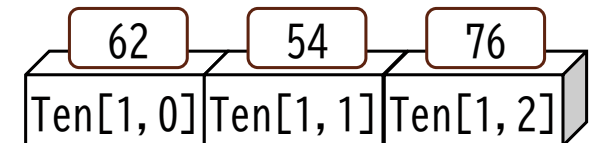
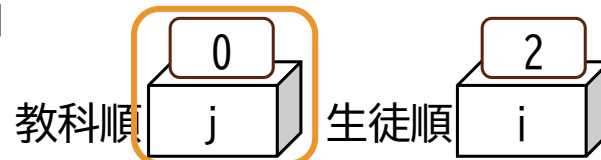
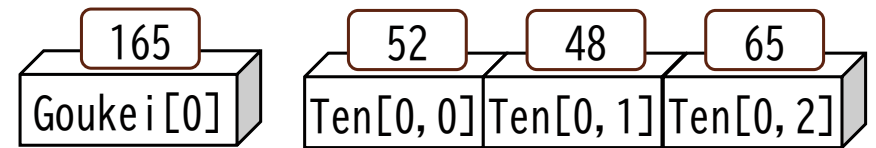
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：…………… 今のjは0
├ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
├   └ Goukei[j] = Goukei[j] + Ten[j, i]
└   表示する(Goukei[j]/3)

```

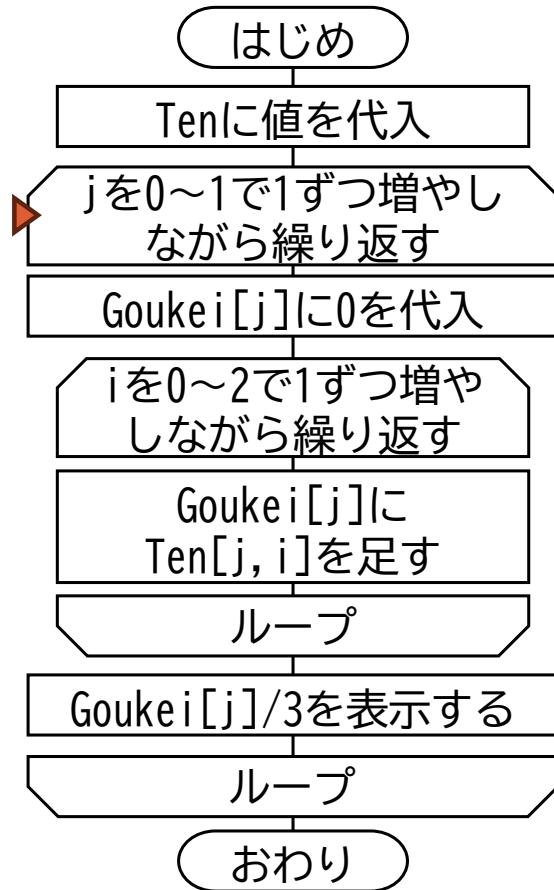
※変数の意味を把握する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

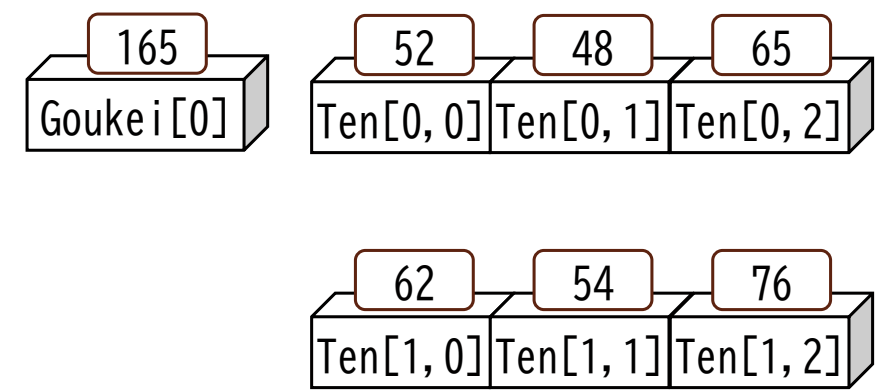
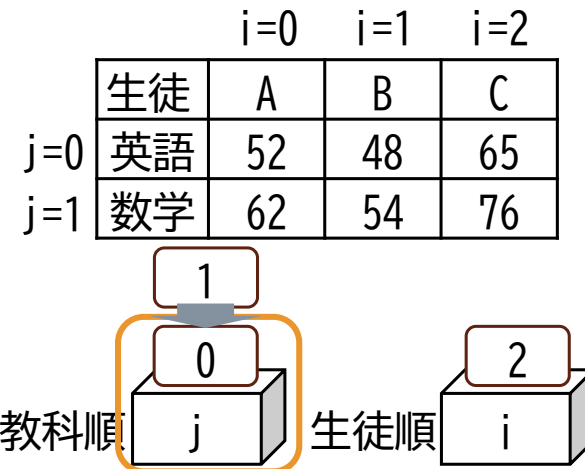


共通テストで 사용되는DNCL

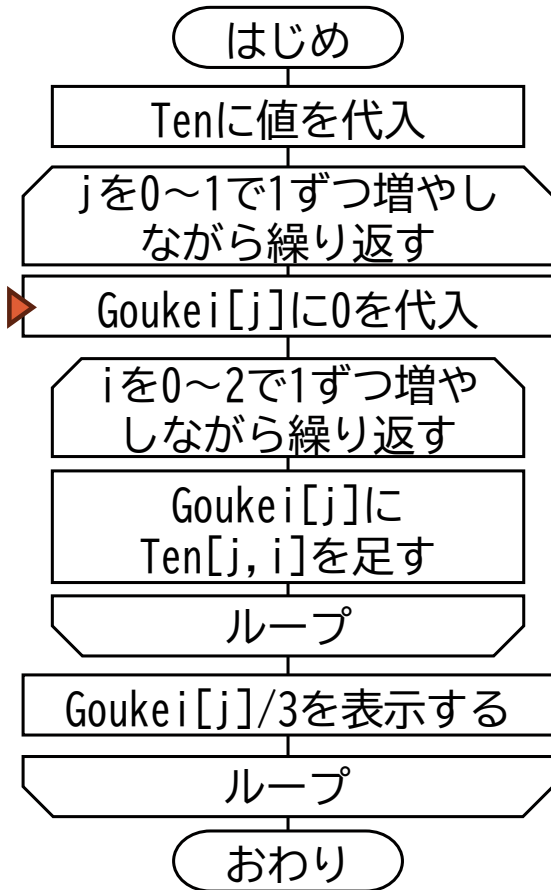
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：…………… 今のjは0
├ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
├   └ Goukei[j] = Goukei[j] + Ten[j, i]
└   表示する(Goukei[j]/3)
    
```

※変数の意味を把握する



変数の意味の把握、入れ子の反復処理がキモ



共通テストで 사용되는DNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

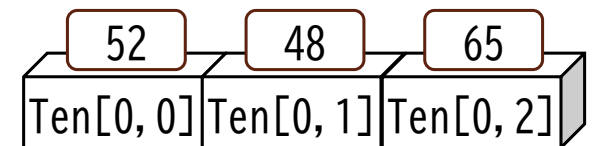
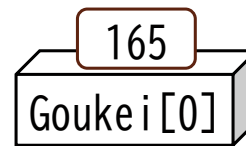
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

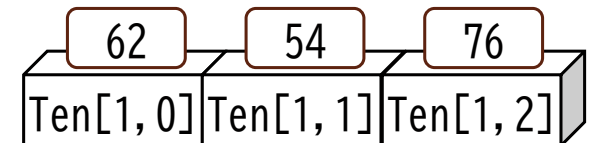
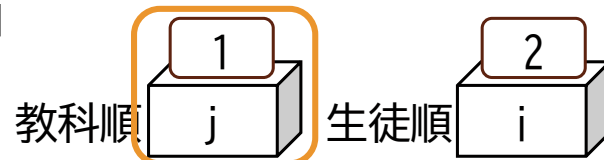
Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



いまのjは1

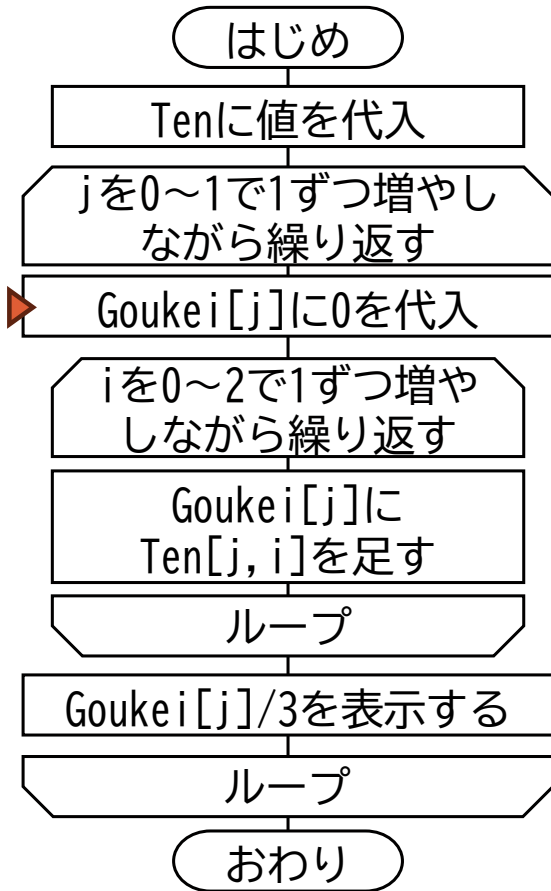


赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える

※変数の意味を把握する

変数の意味の把握、入れ子の反復処理がキモ



共通テストで 사용되는DNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

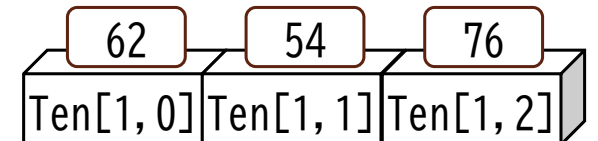
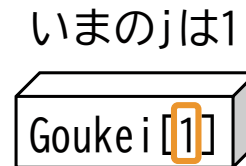
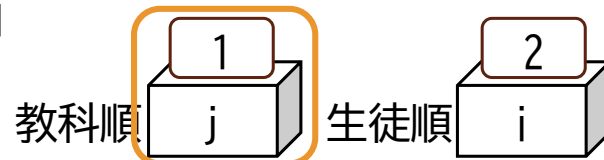
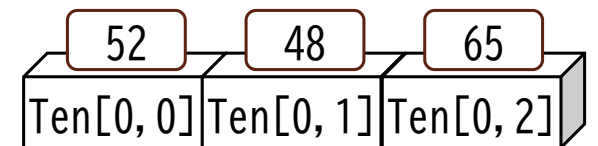
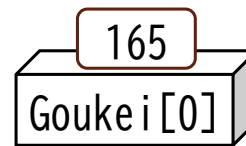
Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

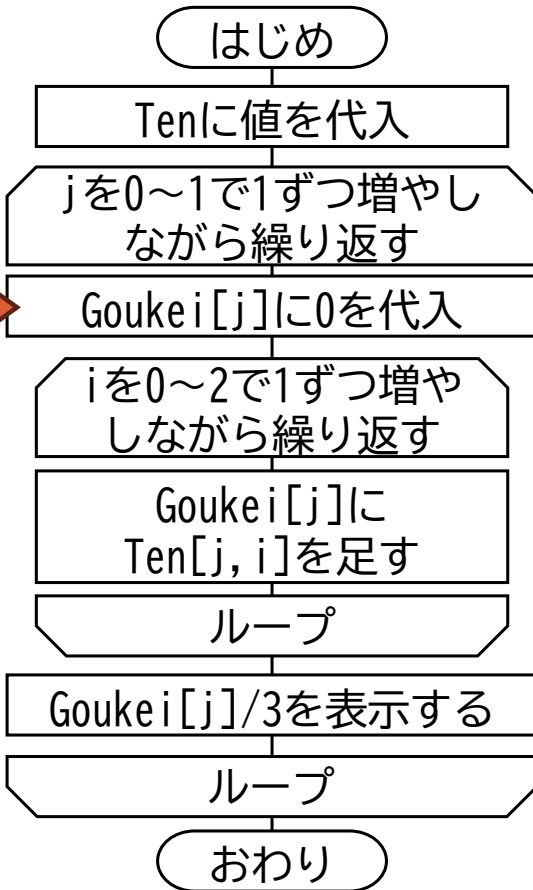
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える

※変数の意味を把握する

変数の意味の把握、入れ子の反復処理がキモ

- ・ 「=」は代入の意味 ※等しいではない
- ・ 計算をしてから変数に代入する
 - ・ 変数は値に置き換える

※変数の意味を把握する



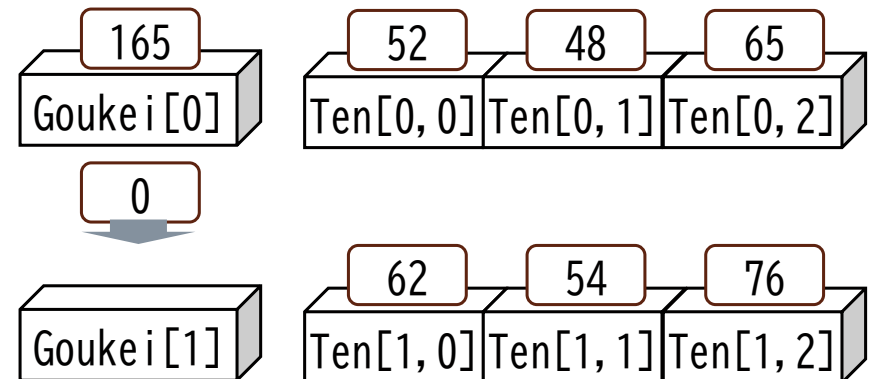
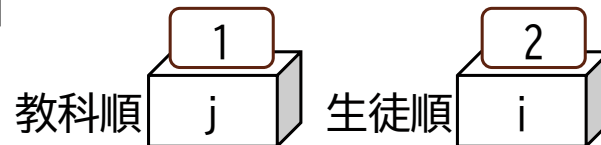
共通テストで使用されるDNCL

```

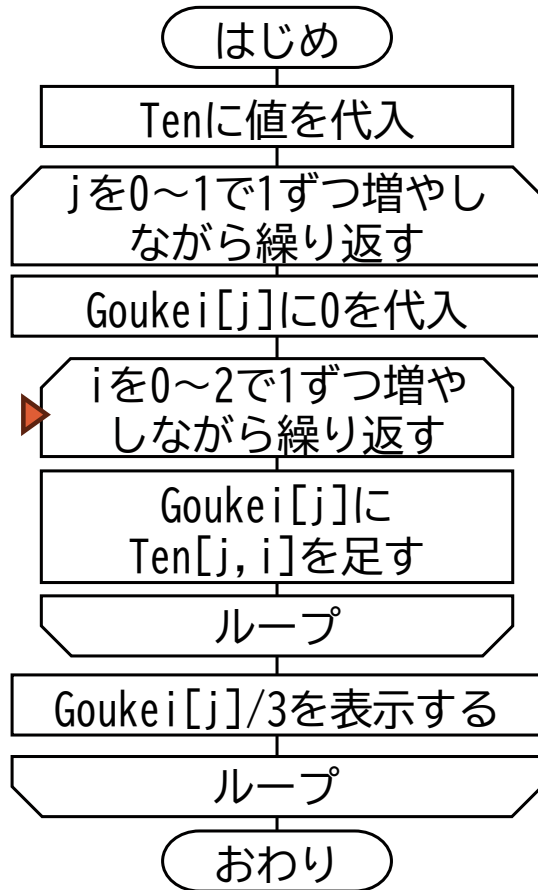
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：……

Goukei[j] = Goukei[j] + Ten[j, i]

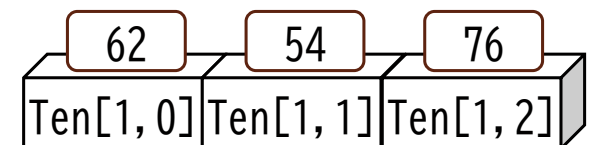
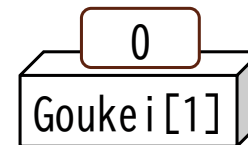
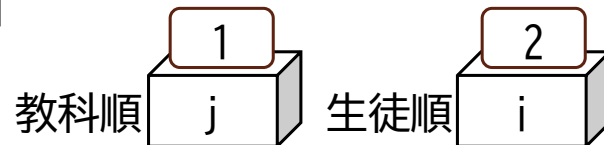
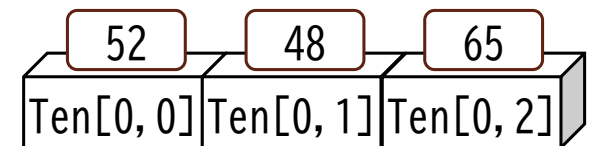
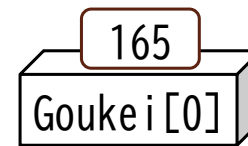
表示する(Goukei[j]/3)

※変数の意味を把握する

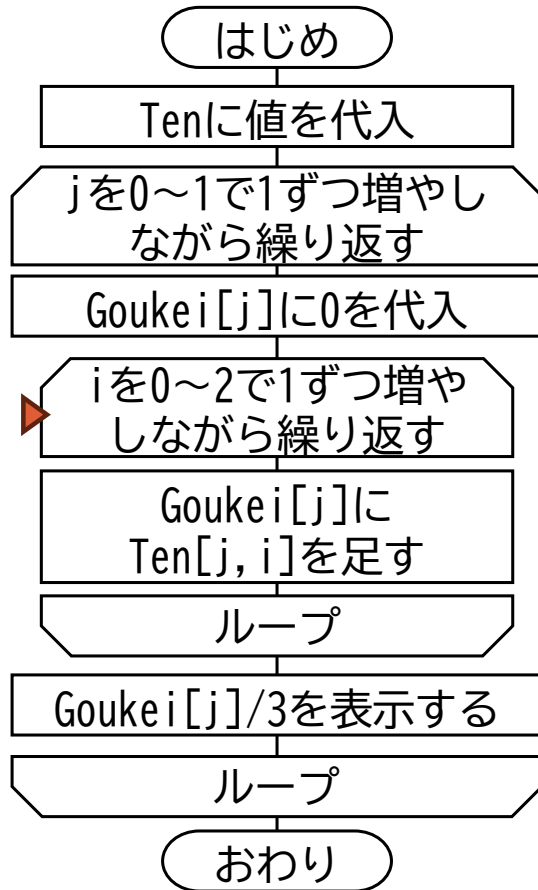
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

手前から反復処理に来た
⇒最初から処理する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

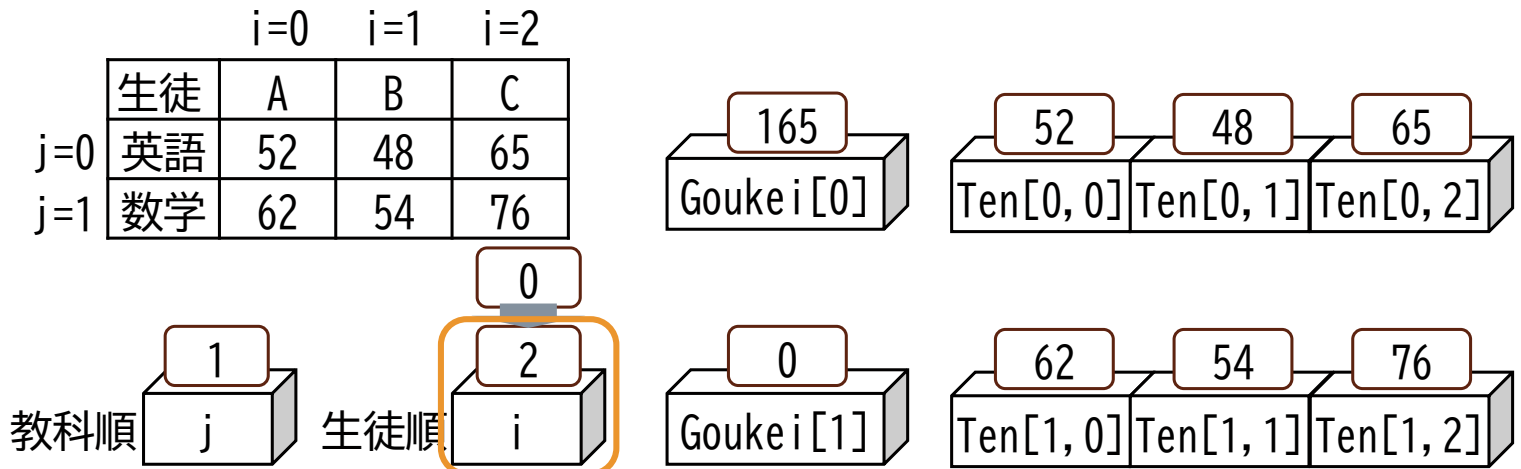
iを0から2まで1ずつ増やしながら繰り返す：……手前から反復処理に来た

Goukei[j] = Goukei[j] + Ten[j, i]

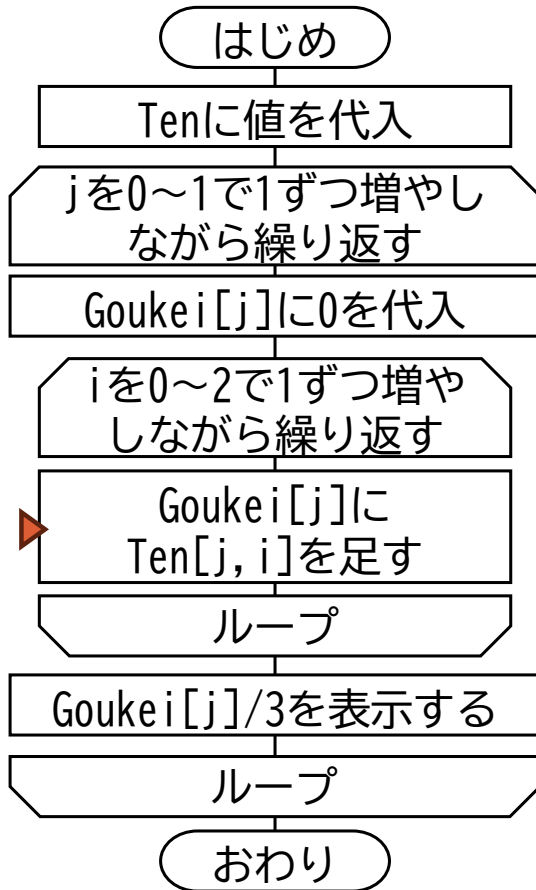
表示する(Goukei[j]/3)

※変数の意味を把握する

⇒最初から処理する



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

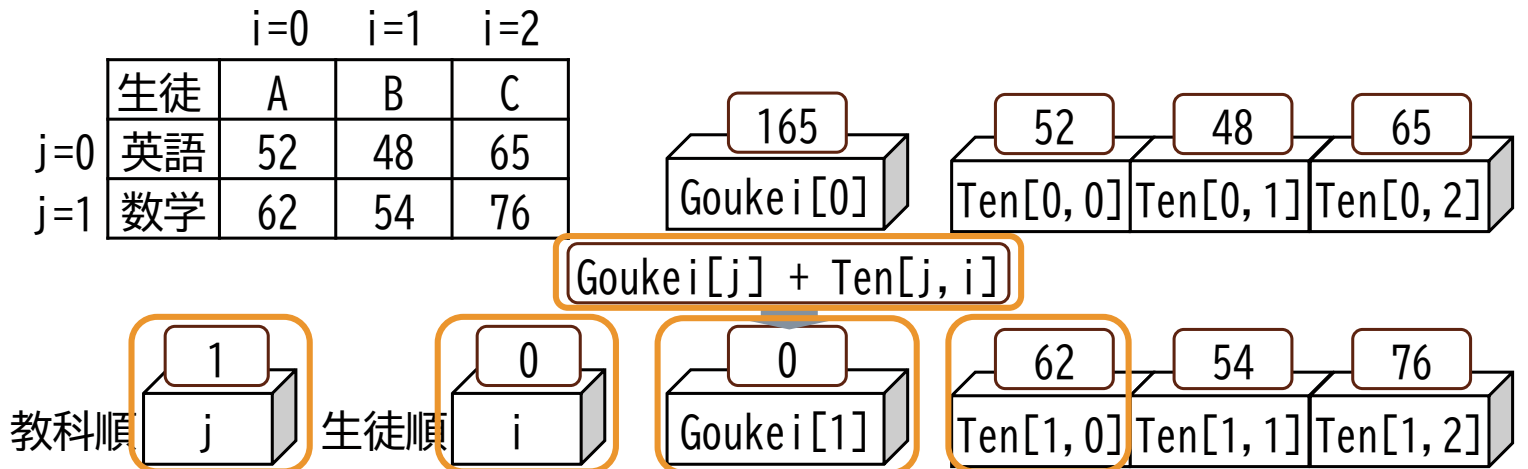
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

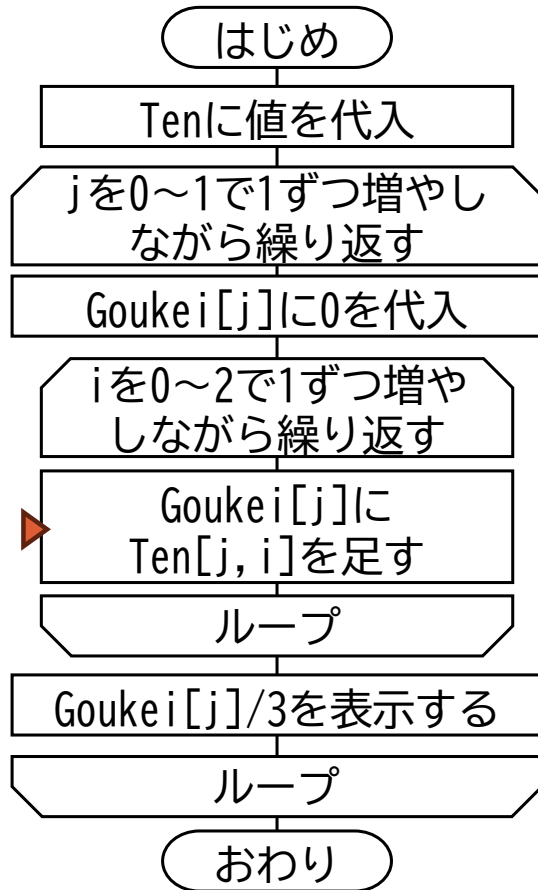
表示する(Goukei[j]/3)

※変数の意味を把握する

いまのjは1、いまのiは0
 いまのGoukei[1]は0
 いまのTen[1, 0]は62



変数の意味の把握、入れ子の反復処理がキモ



共通テストで 사용되는DNCL

Ten = [[52, 48, 65], [62, 54, 76]]

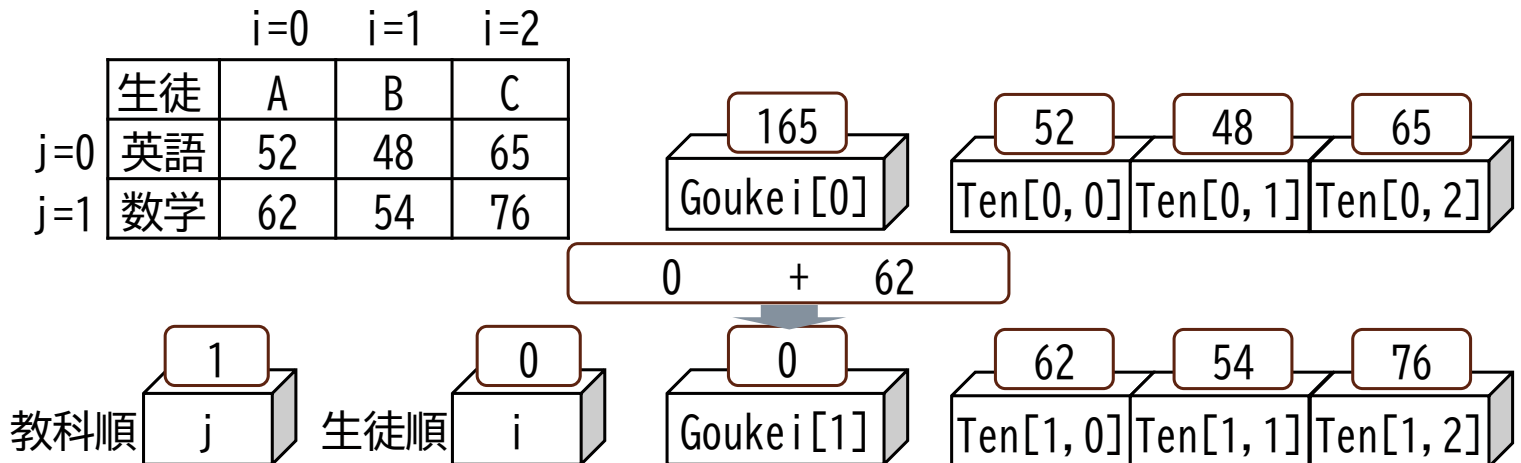
jを0から1まで1ずつ増やしながら繰り返す :

Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す :

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)



※変数の意味を把握する

いまのjは1、いまのiは0
 いまのGoukei[1]は0
 いまのTen[1,0]は62

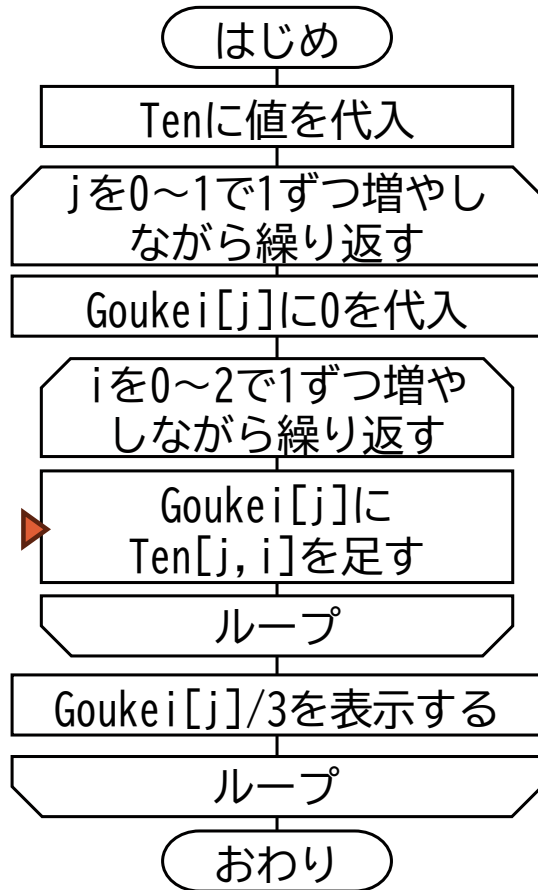
赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

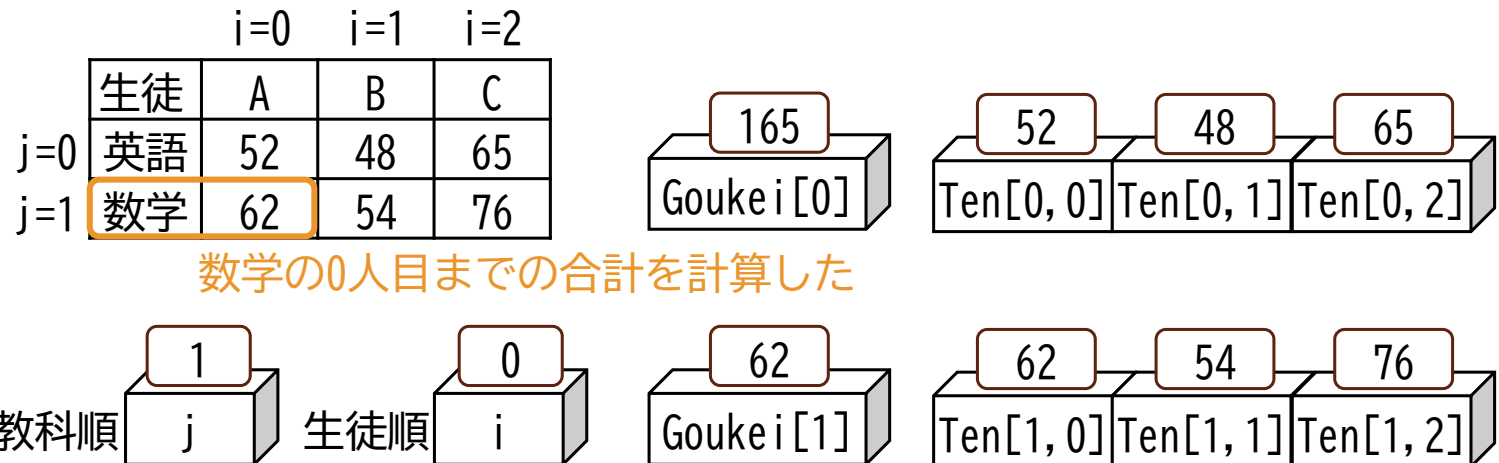
※変数の意味を把握する



共通テストで使用されるDNCL

```

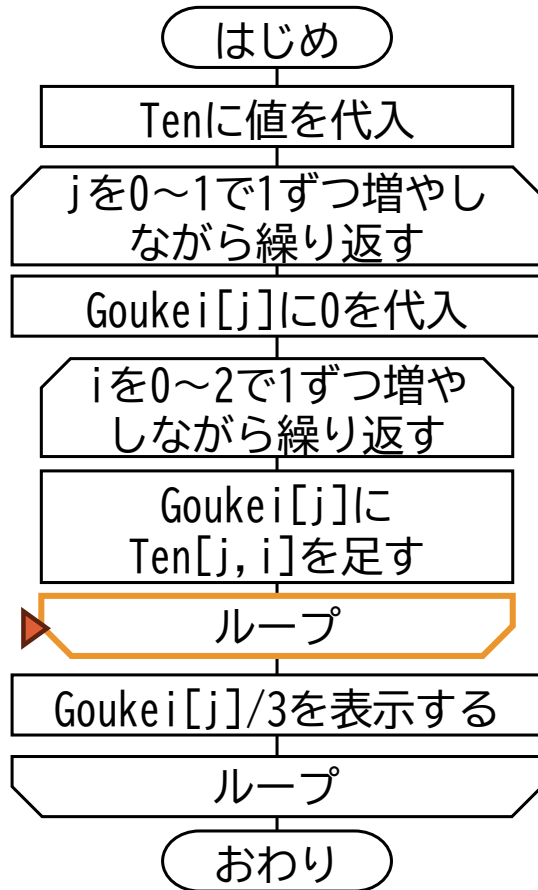
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで使用されるDNCL

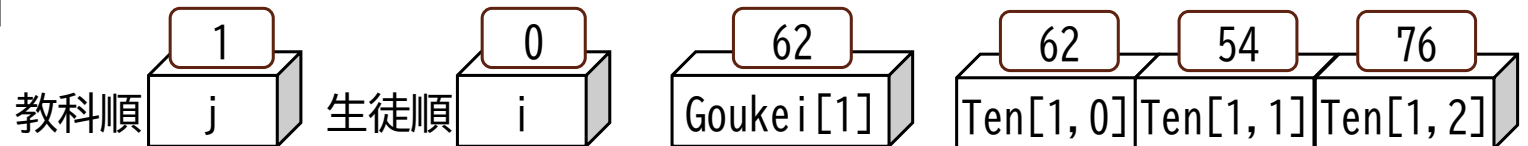
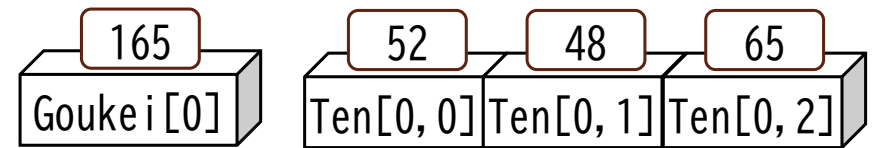
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)

```

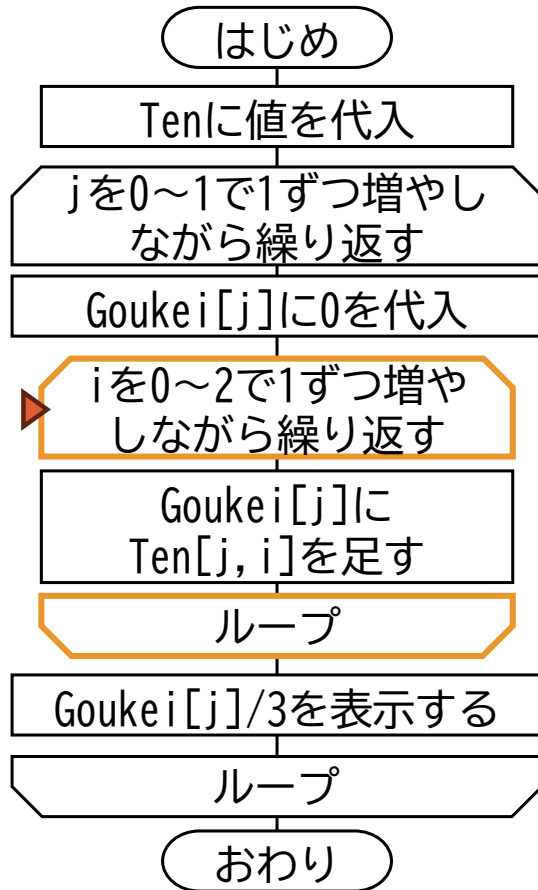
反復処理の終了部分にきたら開始部分に戻る

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで 사용되는DNCL

```

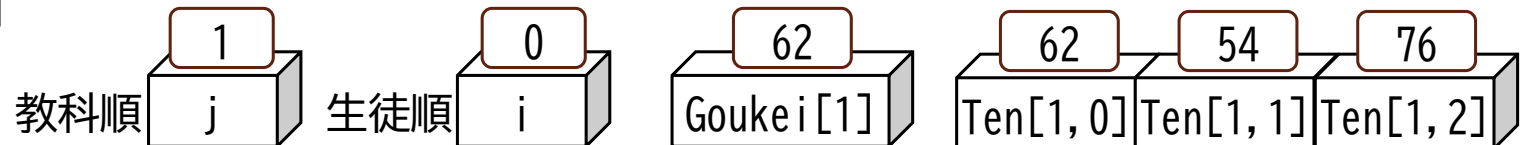
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

※変数の意味を把握する

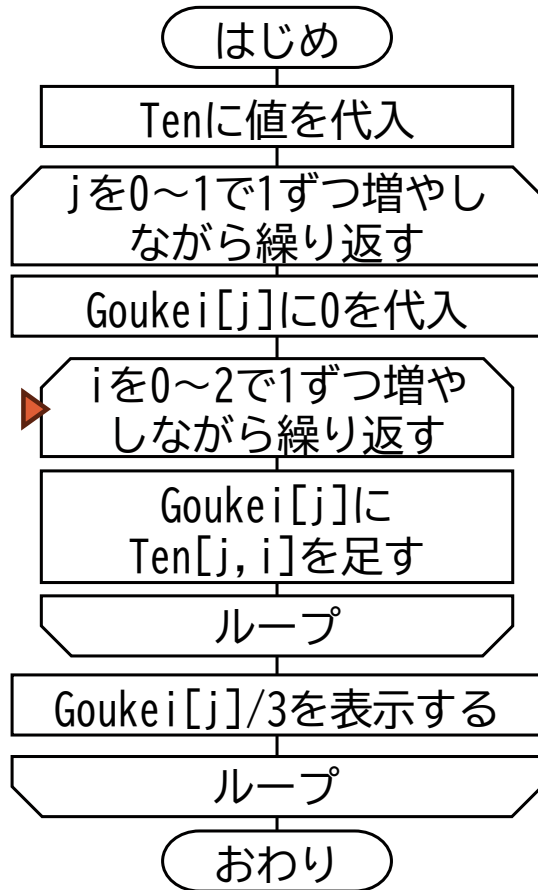
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

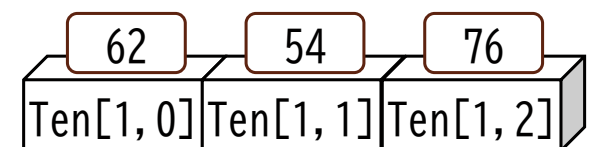
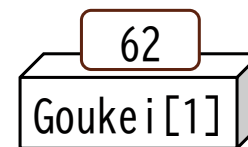
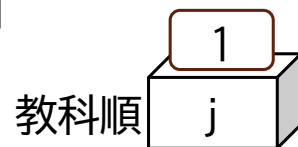
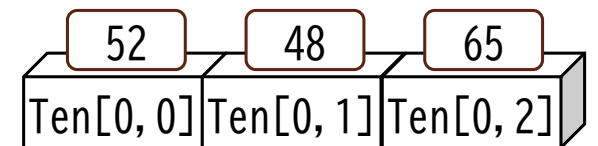
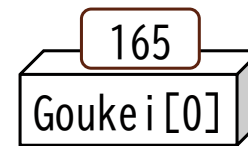
iを0から2まで1ずつ増やしながら繰り返す：………いまのiは0

Goukei[j] = Goukei[j] + Ten[j, i]

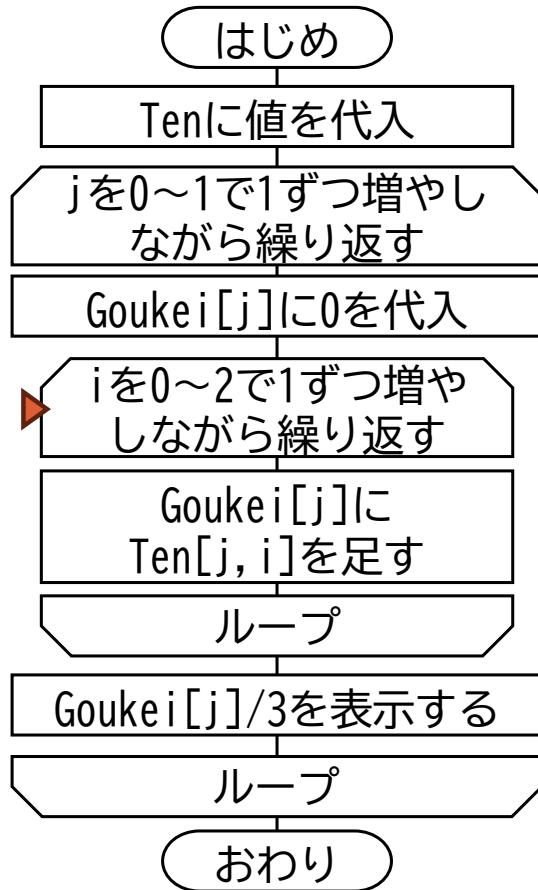
表示する(Goukei[j]/3)

※変数の意味を把握する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

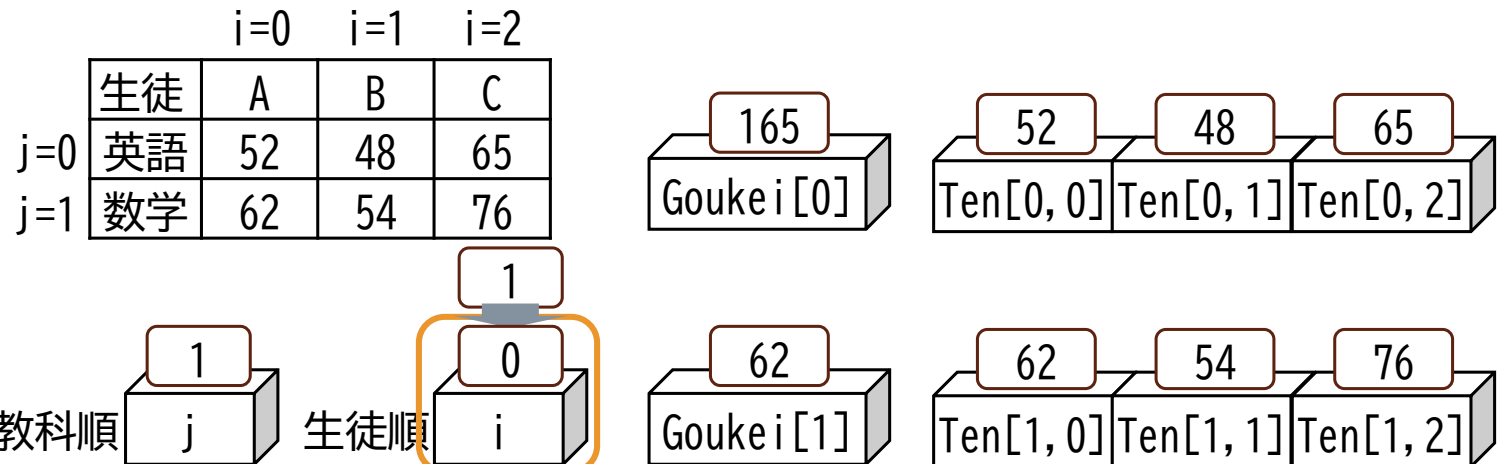
iを0から2まで1ずつ増やしながら繰り返す：………いまのiは0

Goukei[j] = Goukei[j] + Ten[j, i]

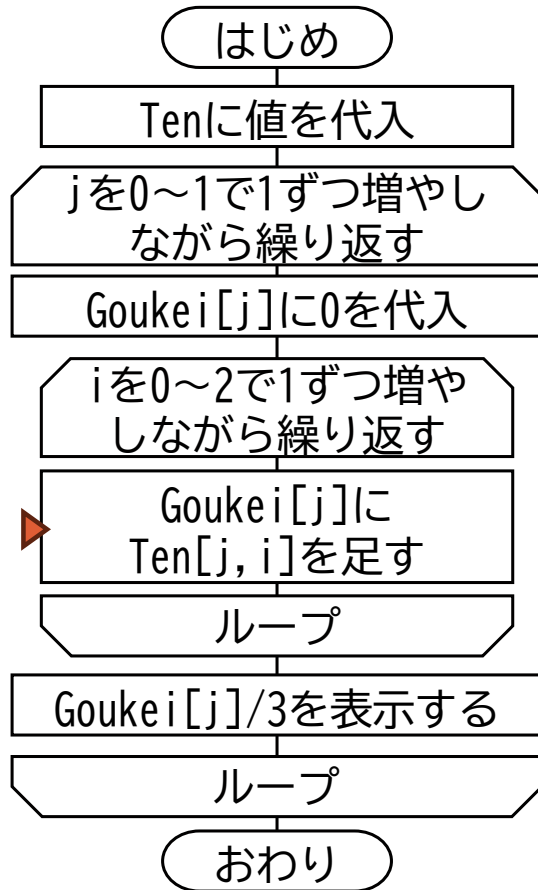
表示する(Goukei[j]/3)

※変数の意味を把握する

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
- 変数は値に置き換える



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

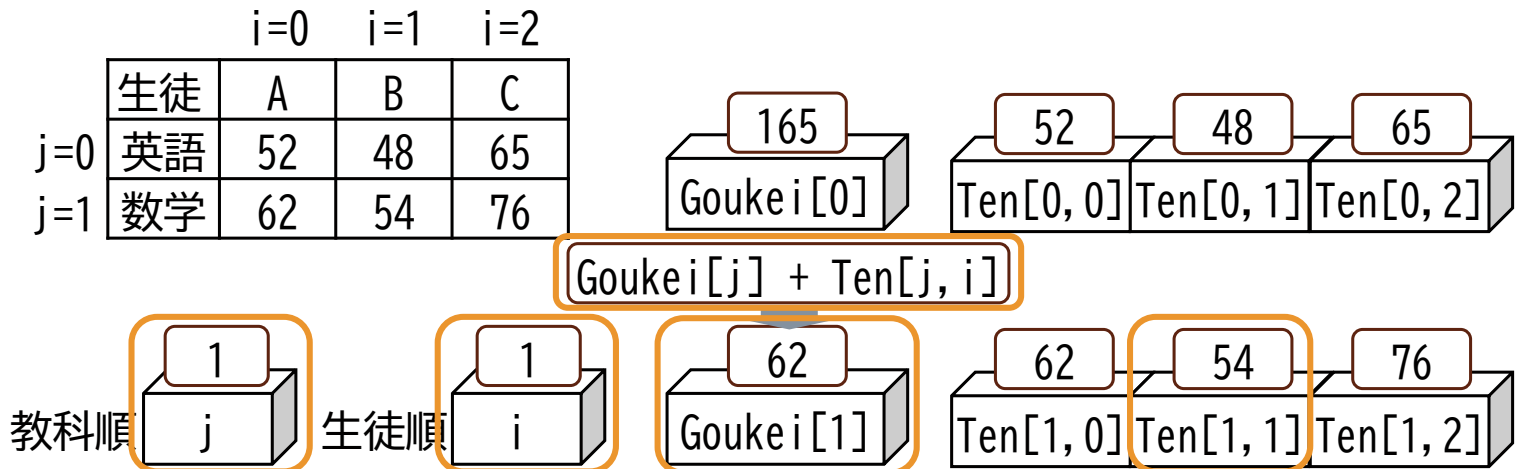
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)

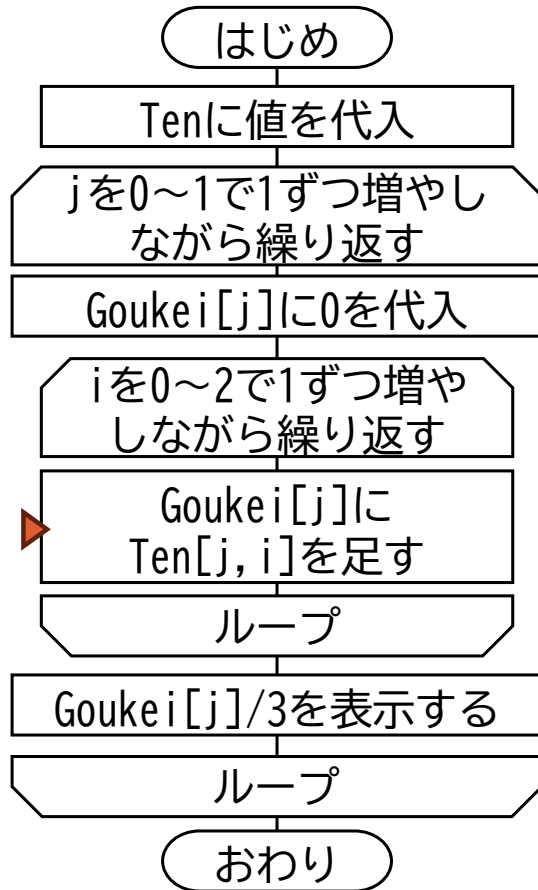
※変数の意味を把握する

いまのjは1、いまのiは1
 いまのGoukei[1]は62
 いまのTen[1, 1]は54



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



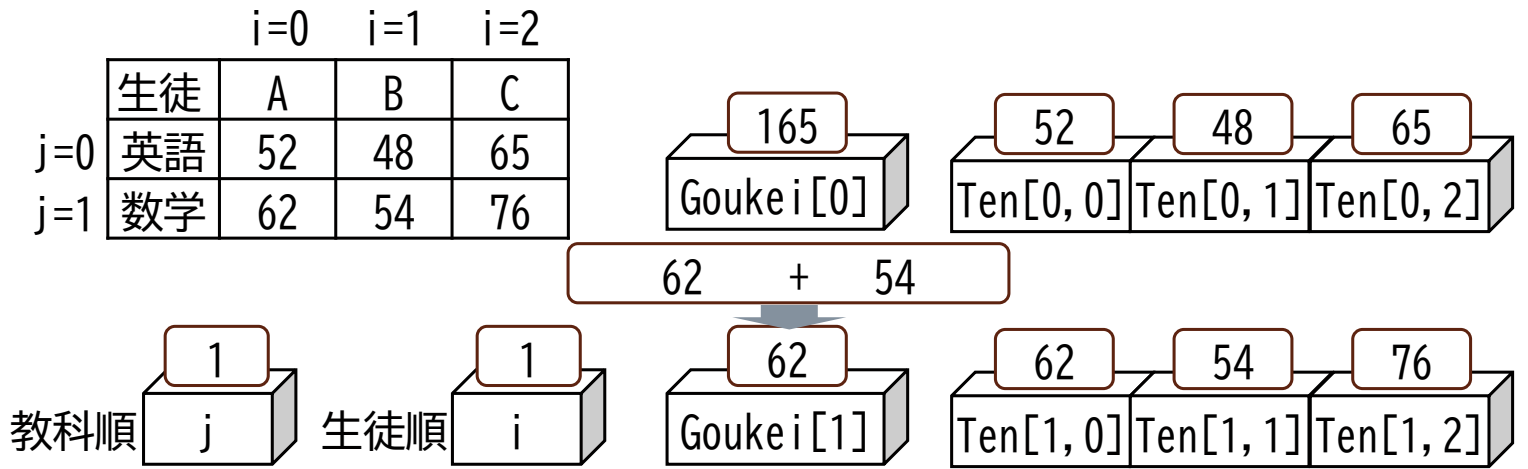
共通テストで使われるDNCL

```

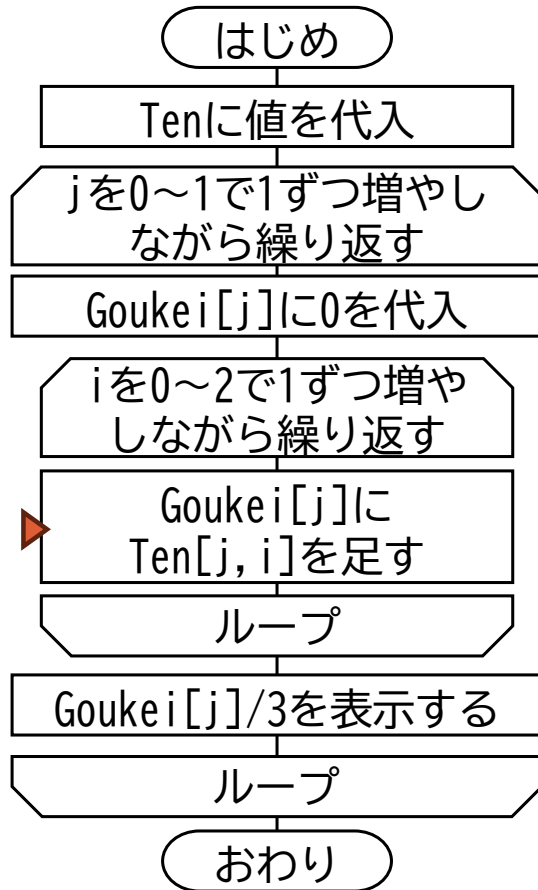
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながらか繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながらか繰り返す：
└ ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

いまのjは1、いまのiは1
 いまのGoukei[1]は62
 いまのTen[1, 1]は54



変数の意味の把握、入れ子の反復処理がキモ



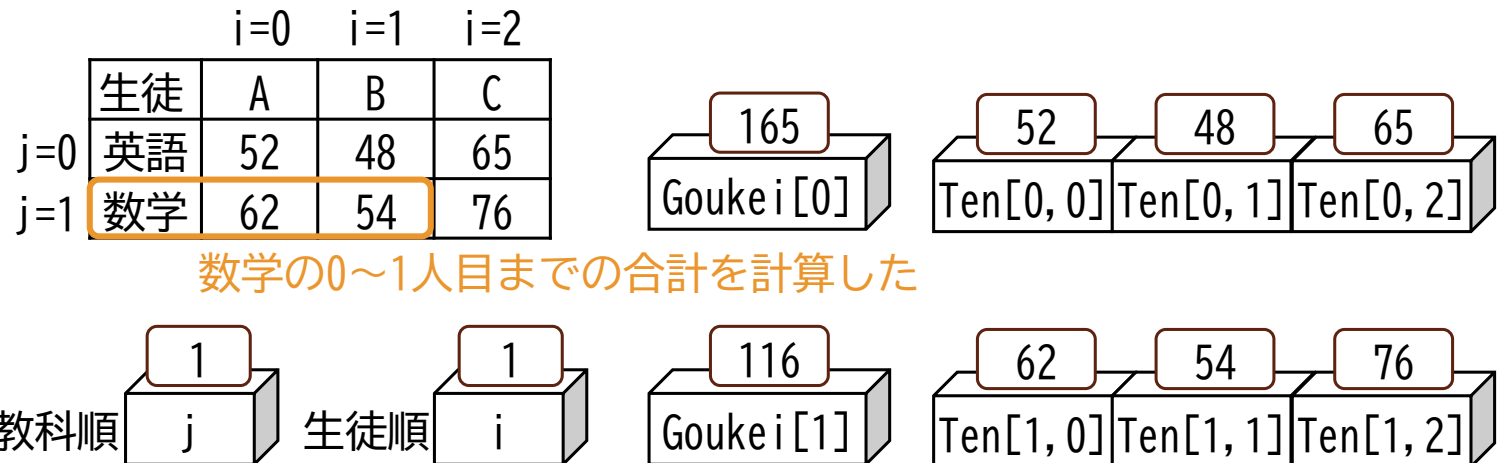
共通テストで使われるDNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└┬ Goukei[j] = Goukei[j] + Ten[j, i]
  └ 表示する(Goukei[j]/3)
  
```

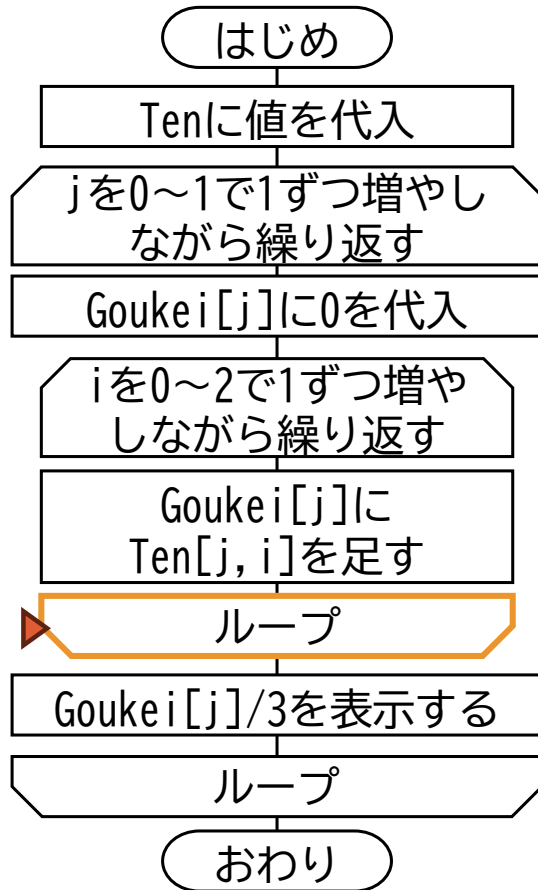
※変数の意味を把握する

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使用されるDNCL

```

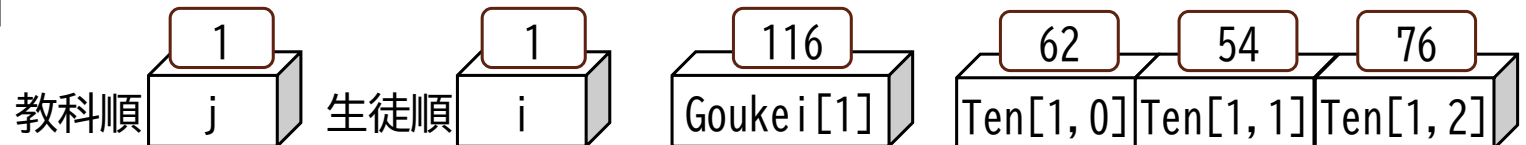
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)

```

※変数の意味を把握する

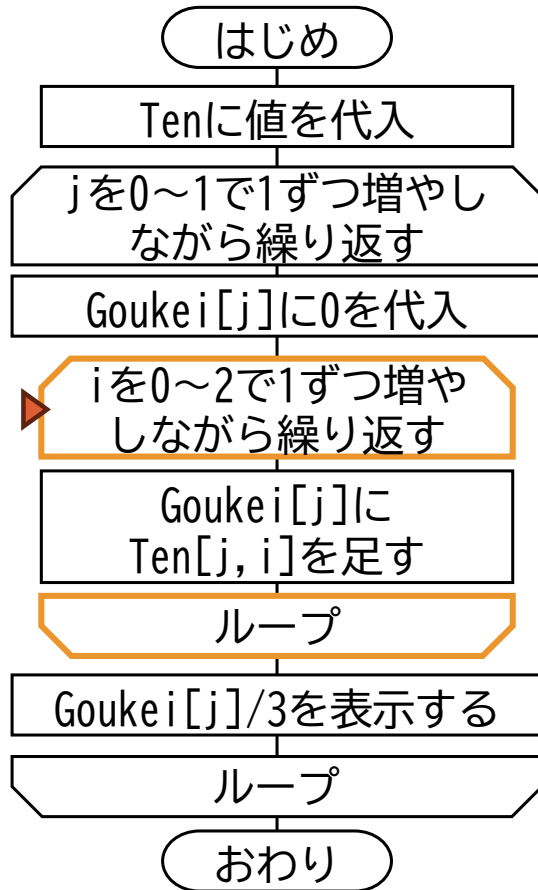
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで 사용되는DNCL

```

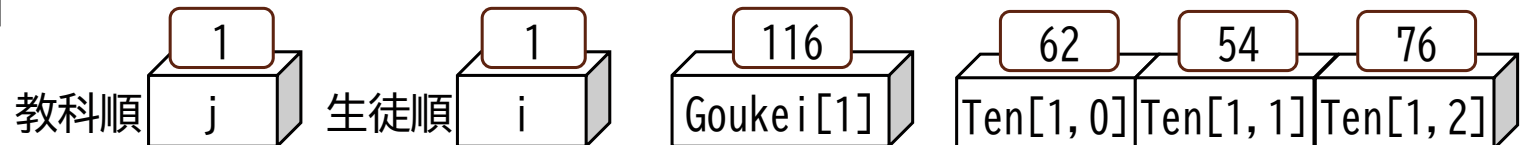
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

※変数の意味を把握する

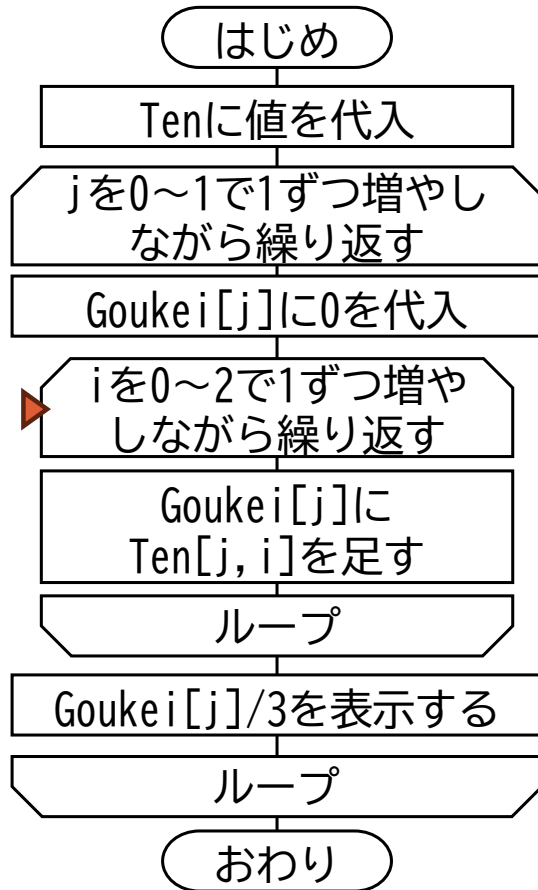
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

反復処理の終了部分にきたら開始部分に戻る



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

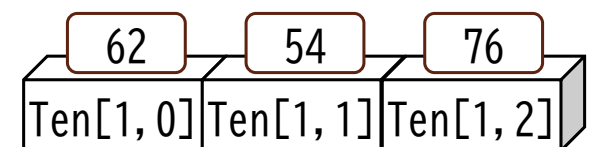
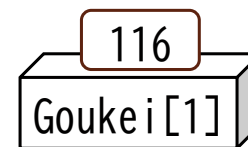
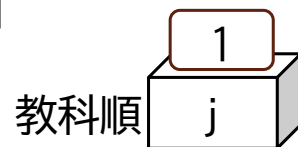
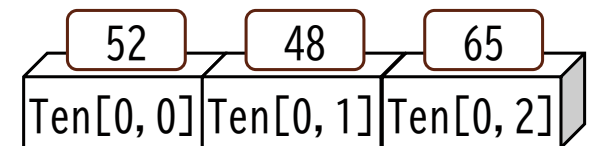
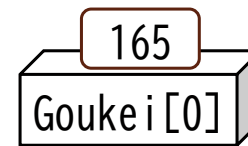
iを0から2まで1ずつ増やしながら繰り返す：………いまのiは1

Goukei[j] = Goukei[j] + Ten[j, i]

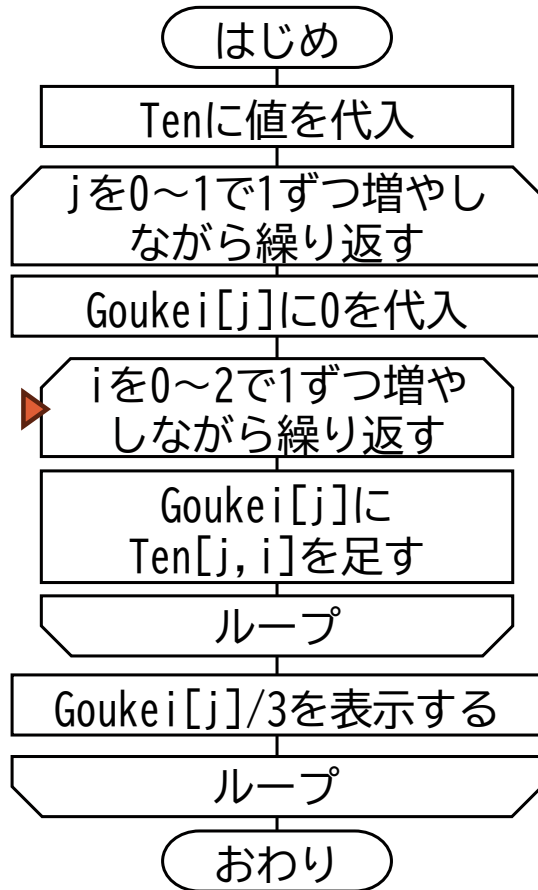
表示する(Goukei[j]/3)

※変数の意味を把握する

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

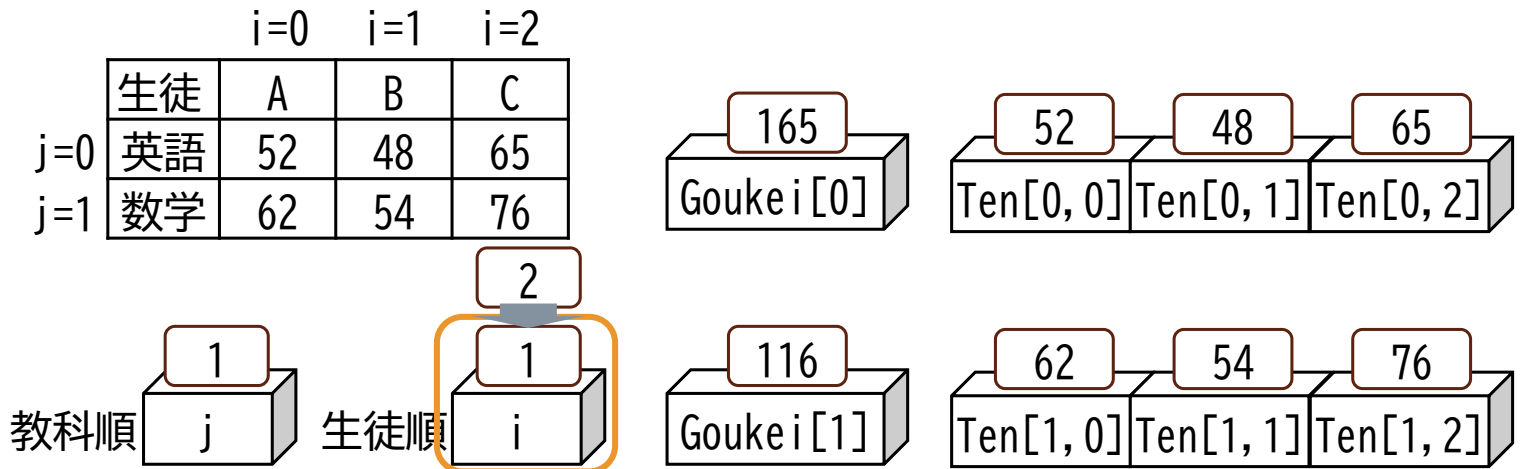
iを0から2まで1ずつ増やしながら繰り返す：………いまのiは1

Goukei[j] = Goukei[j] + Ten[j, i]

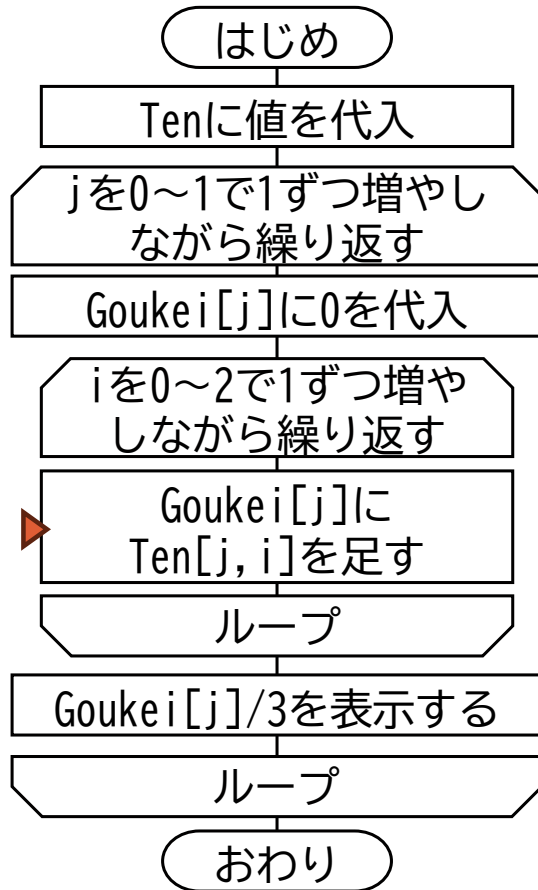
表示する(Goukei[j]/3)

※変数の意味を把握する

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

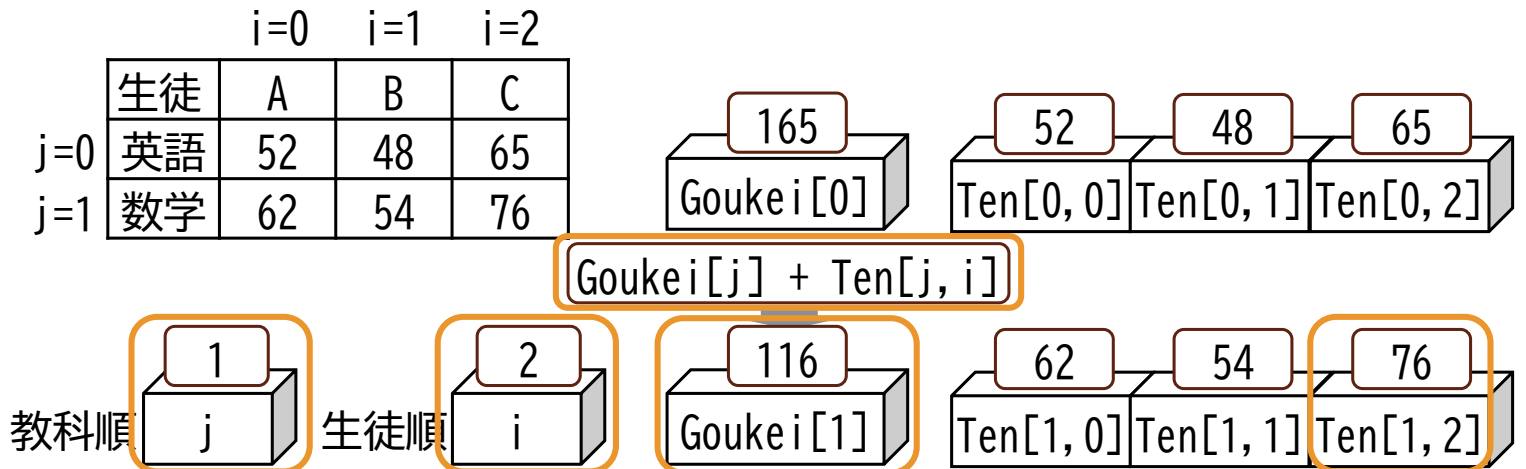
iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

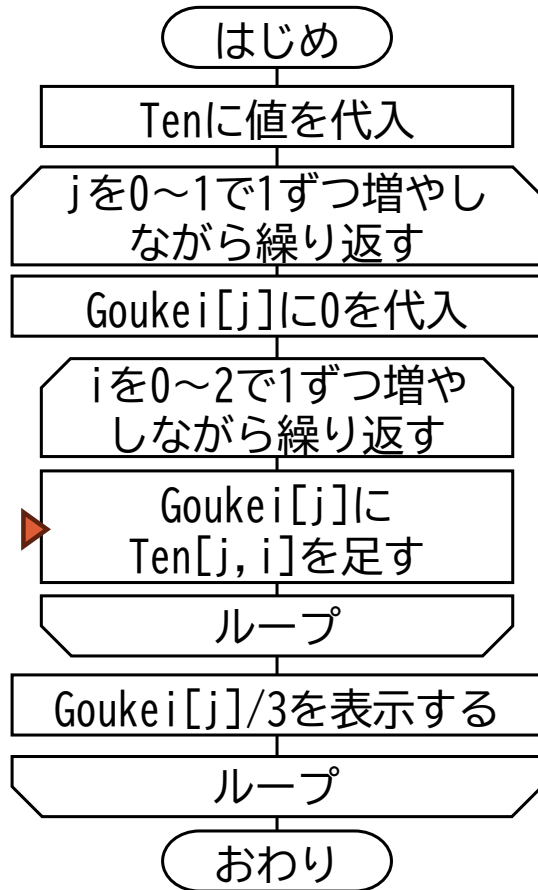
表示する(Goukei[j]/3)

※変数の意味を把握する

いまのjは1、いまのiは2
いまのGoukei[1]は116
いまのTen[1, 2]は76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使われるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

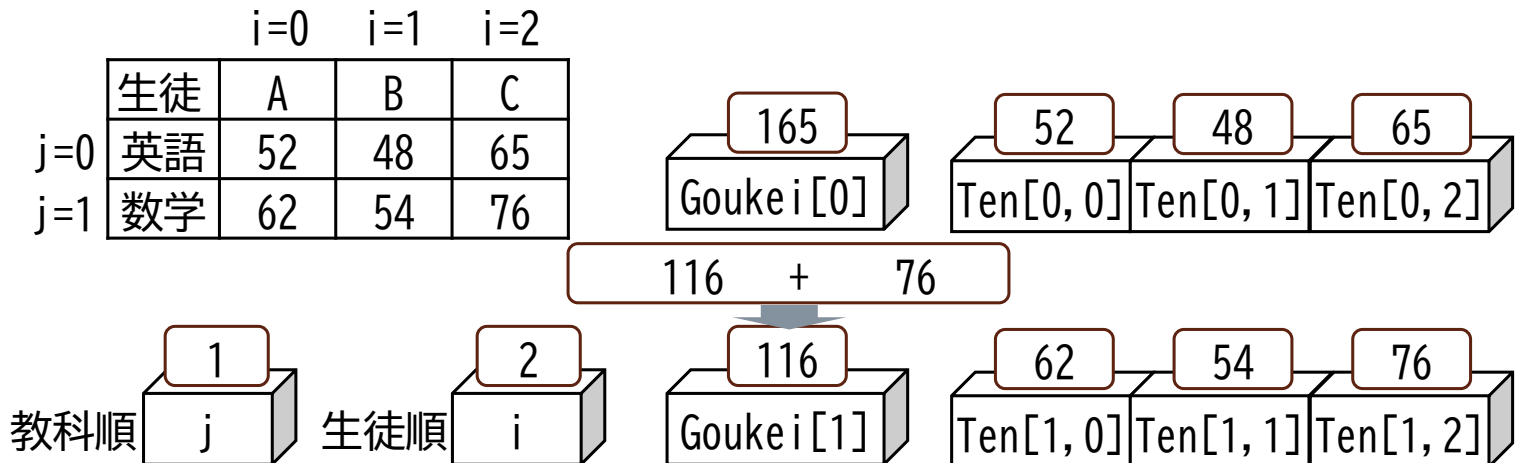
jを0から1まで1ずつ増やしながら繰り返す：

Goukei[j] = 0

iを0から2まで1ずつ増やしながら繰り返す：

Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)



赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

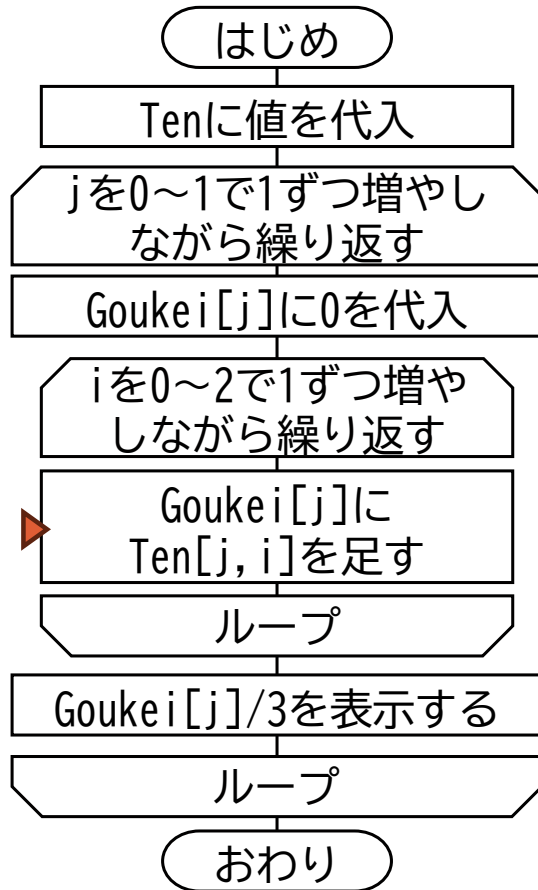
※変数の意味を把握する

いまのjは1、いまのiは2
 いまのGoukei[1]は116
 いまのTen[1,2]は76

変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

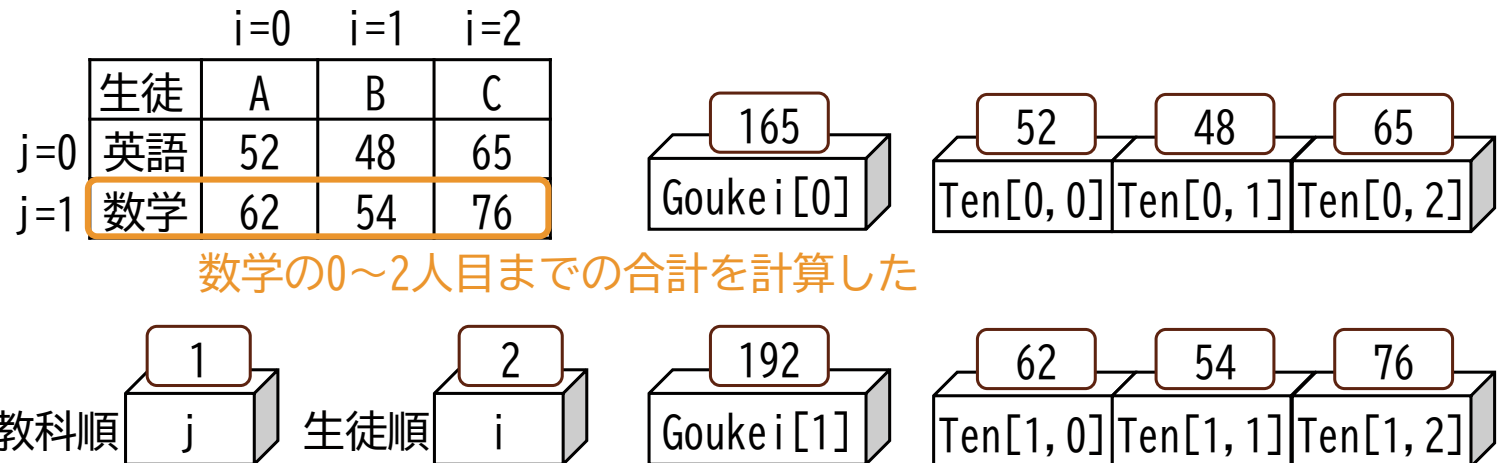
※変数の意味を把握する



共通テストで使用されるDNCL

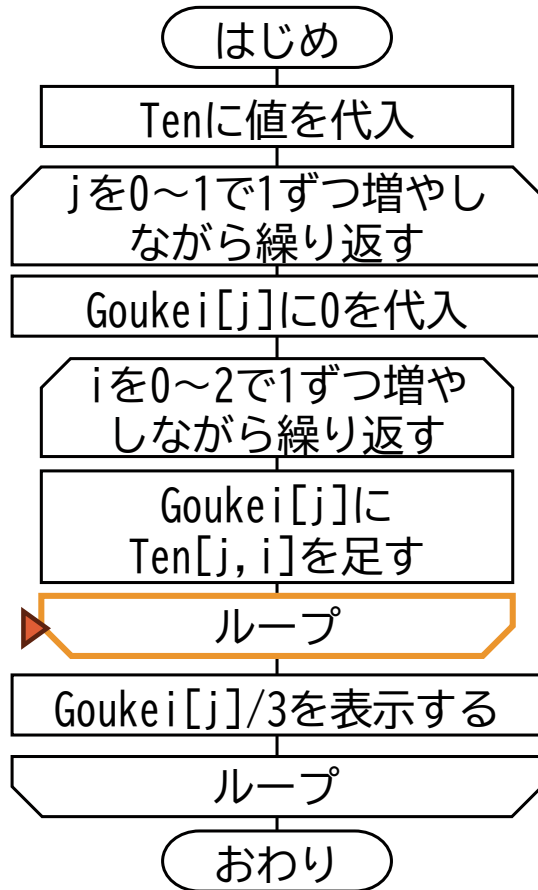
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながらか繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながらか繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使用されるDNCL

```

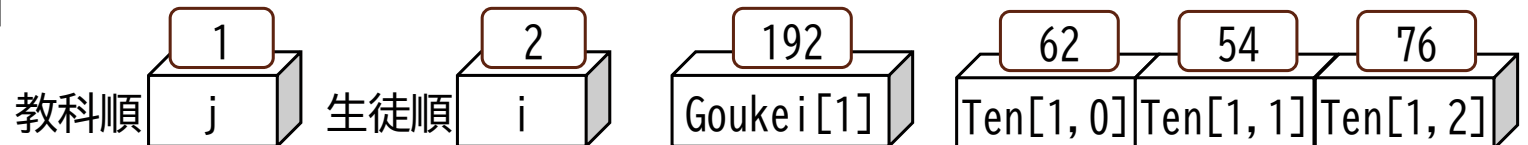
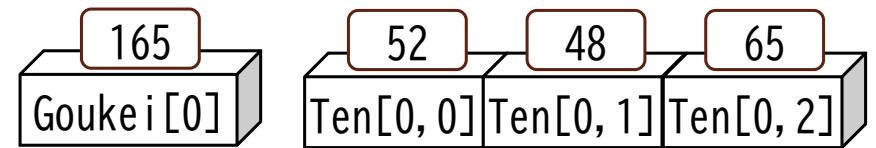
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```

※変数の意味を把握する

反復処理の終了部分にきたら開始部分に戻る

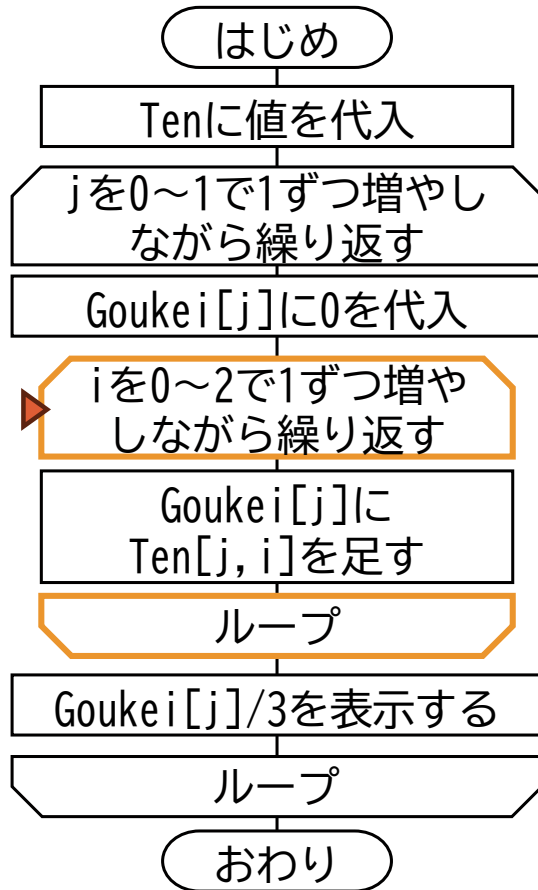
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで 사용되는DNCL

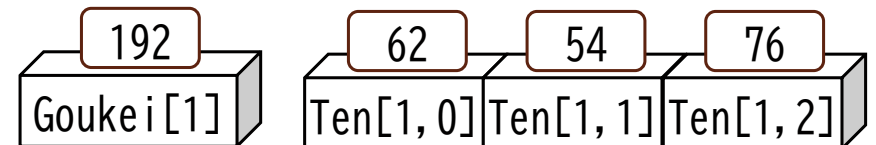
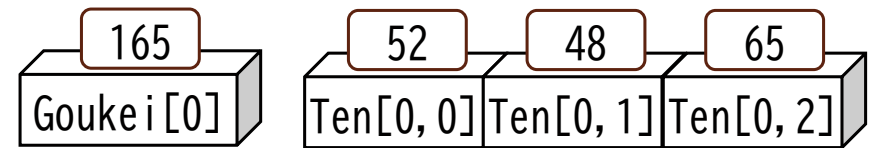
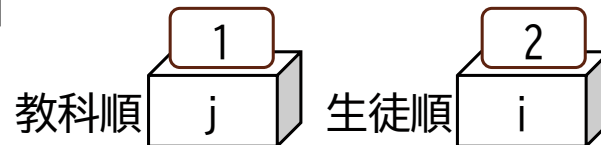
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

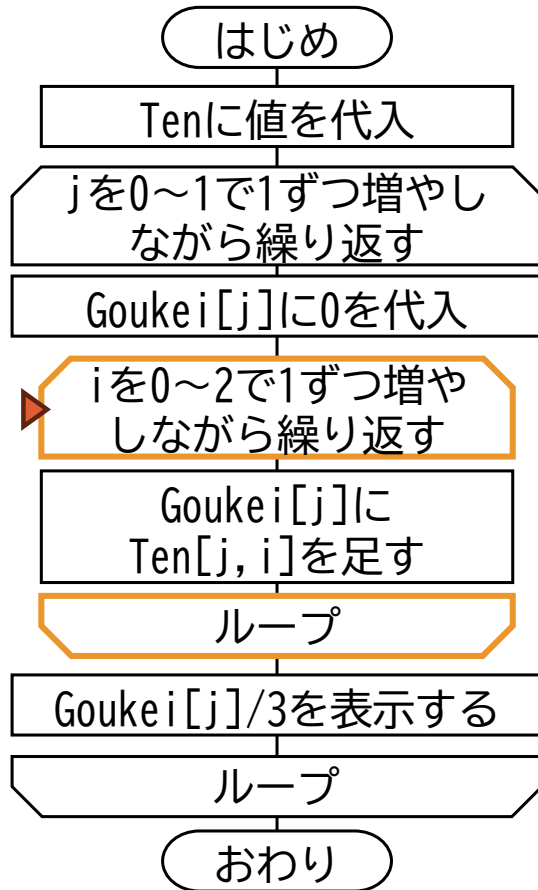
```

反復処理の終了部分にきたら開始部分に戻る

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで 사용되는DNCL

```

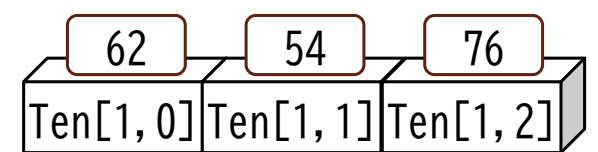
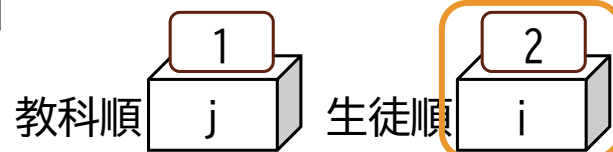
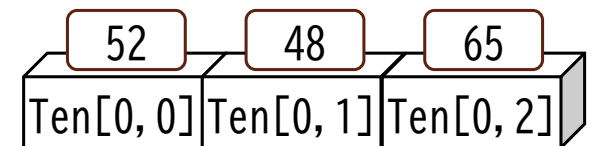
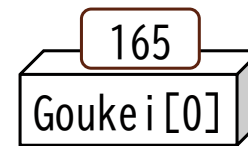
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：………いまのiは2
  Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

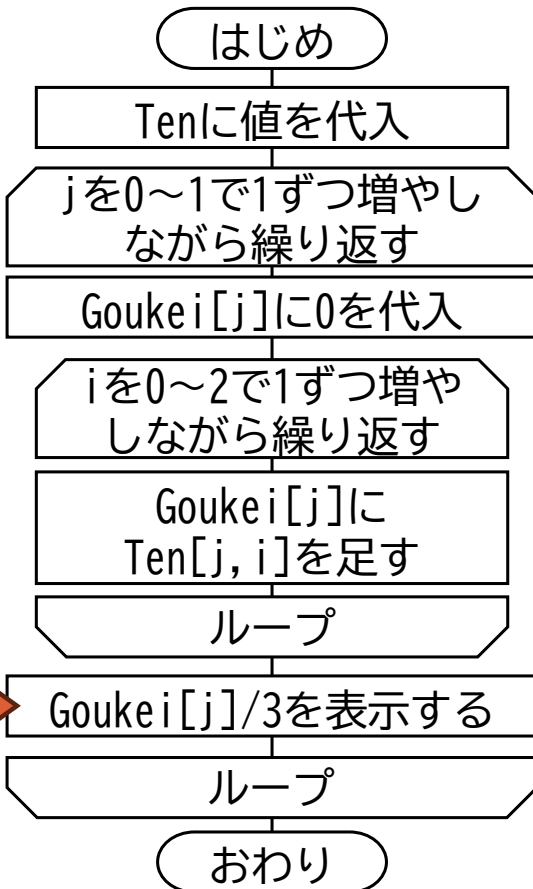
上限に達したので反復処理の次に進む

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ

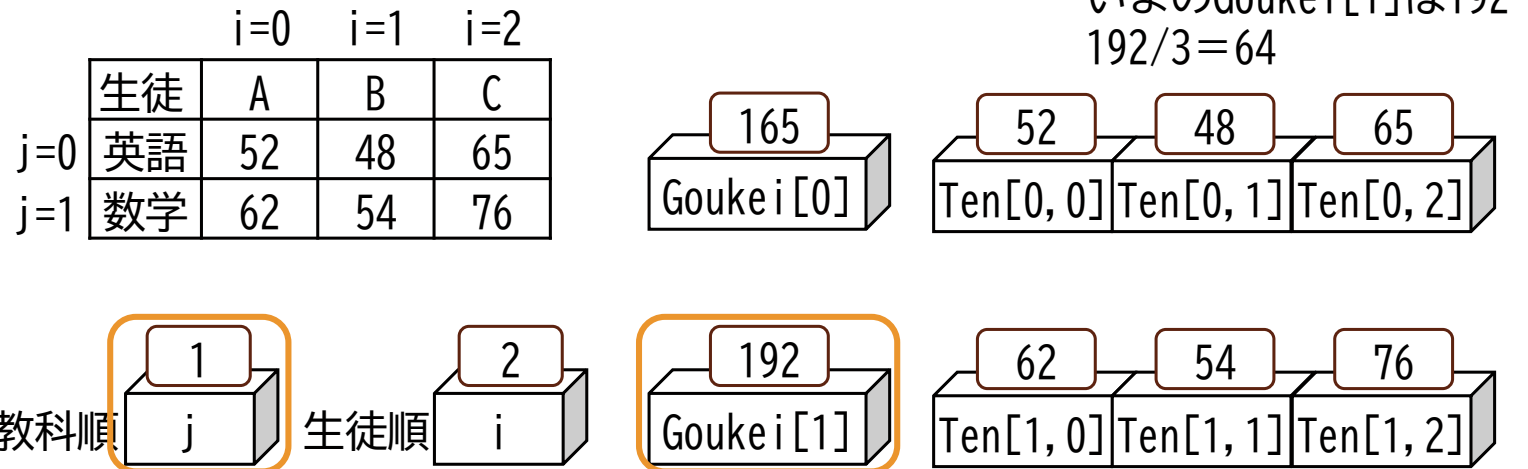
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



共通テストで使用されるDNCL

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
    Goukei[j] = 0
    iを0から2まで1ずつ増やしながら繰り返す：
        Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)
```

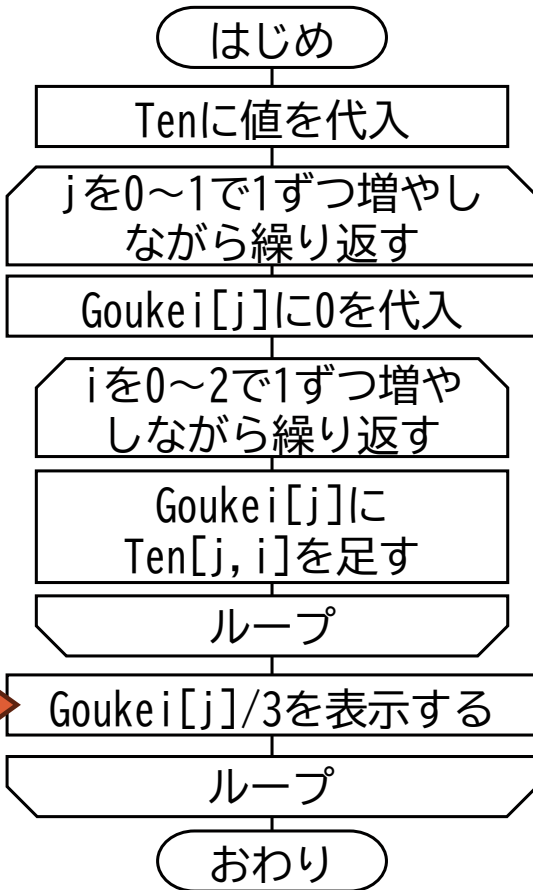
※変数の意味を把握する



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する



共通テストで使用されるDNCL

Ten = [[52, 48, 65], [62, 54, 76]]

jを0から1まで1ずつ増やしなが

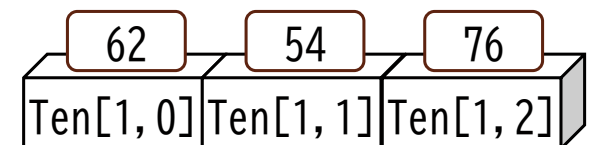
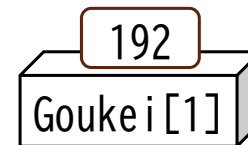
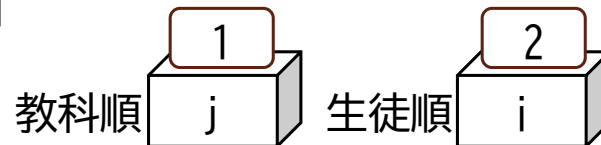
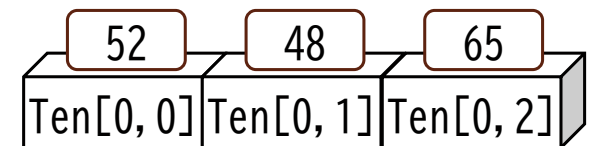
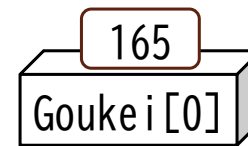
ら繰り返す：
Goukei[j] = 0

iを0から2まで1ずつ増やしなが

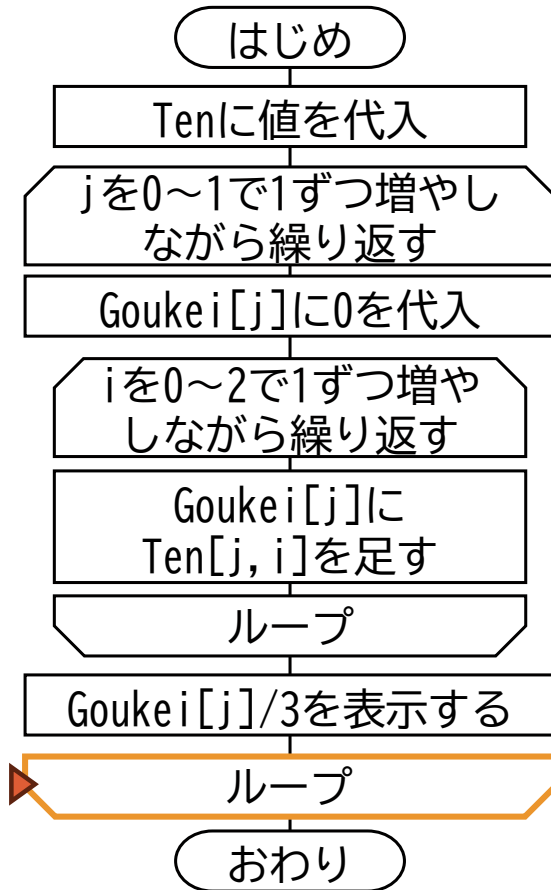
ら繰り返す：
Goukei[j] = Goukei[j] + Ten[j, i]

表示する(Goukei[j]/3)..... 「64」と表示される

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

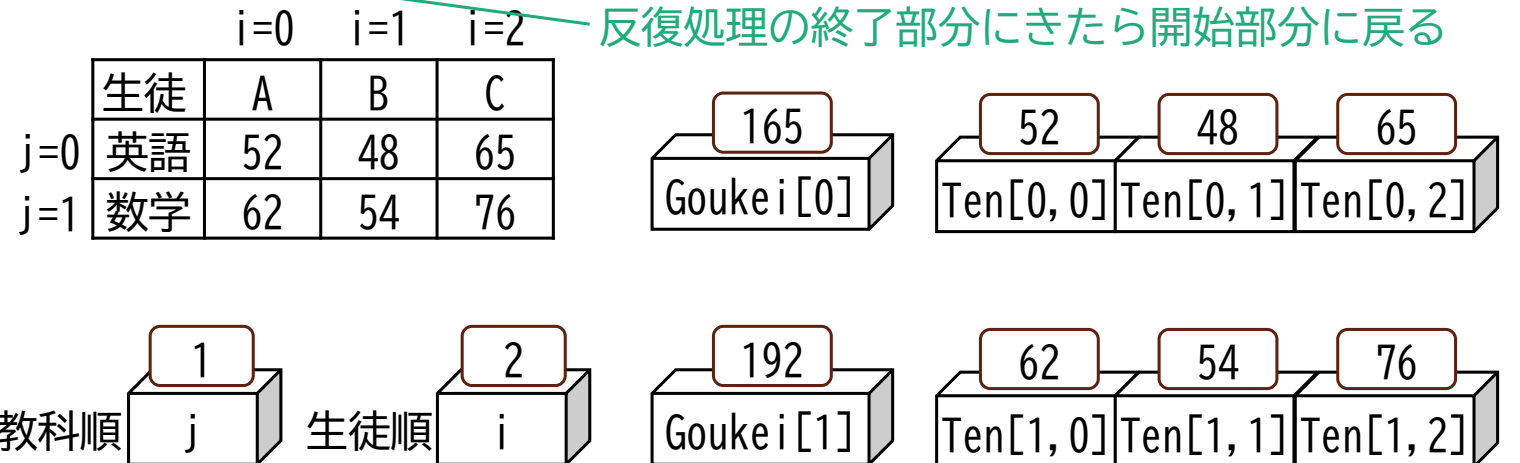
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
    Goukei[j] = 0
    iを0から2まで1ずつ増やしながら繰り返す：
        Goukei[j] = Goukei[j] + Ten[j, i]
    表示する(Goukei[j]/3)
    
```

※変数の意味を把握する

赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

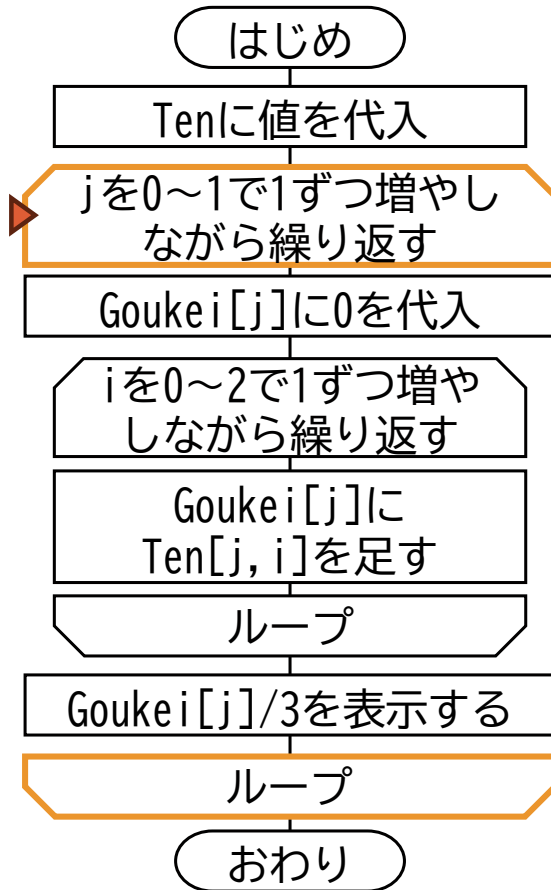
- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える



変数の意味の把握、入れ子の反復処理がキモ

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

※変数の意味を把握する

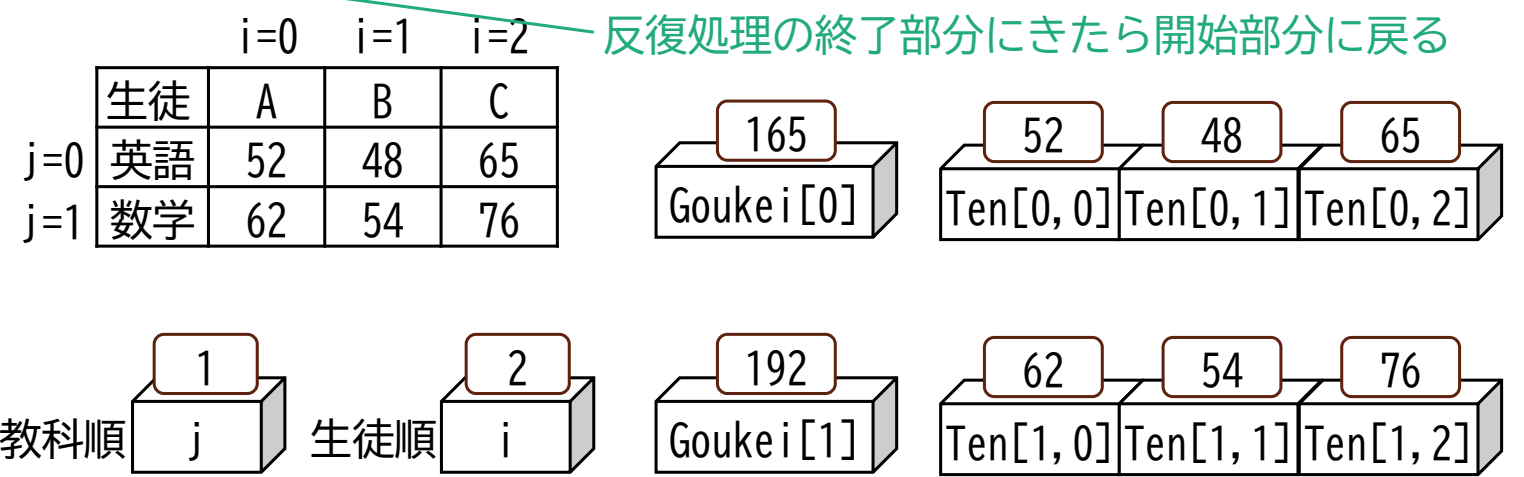


共通テストで 사용되는DNCL

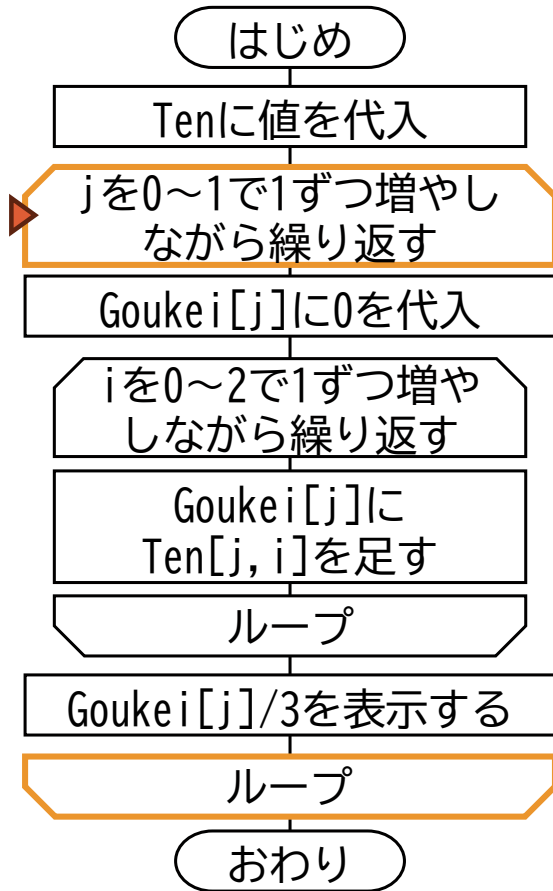
```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)

```



変数の意味の把握、入れ子の反復処理がキモ



共通テストで使用されるDNCL

```

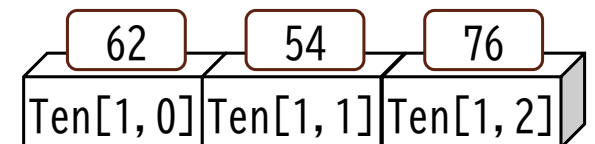
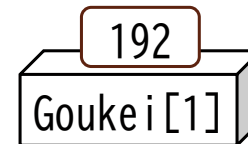
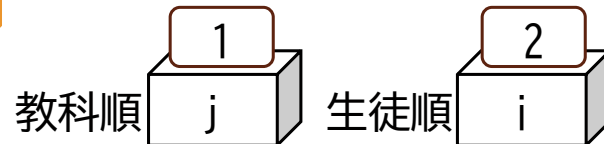
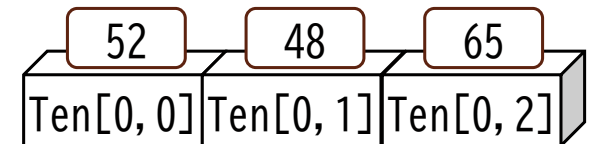
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す : .....
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながら繰り返す :
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
    
```

※変数の意味を把握する
 上限に達したので
 反復処理の次に進む

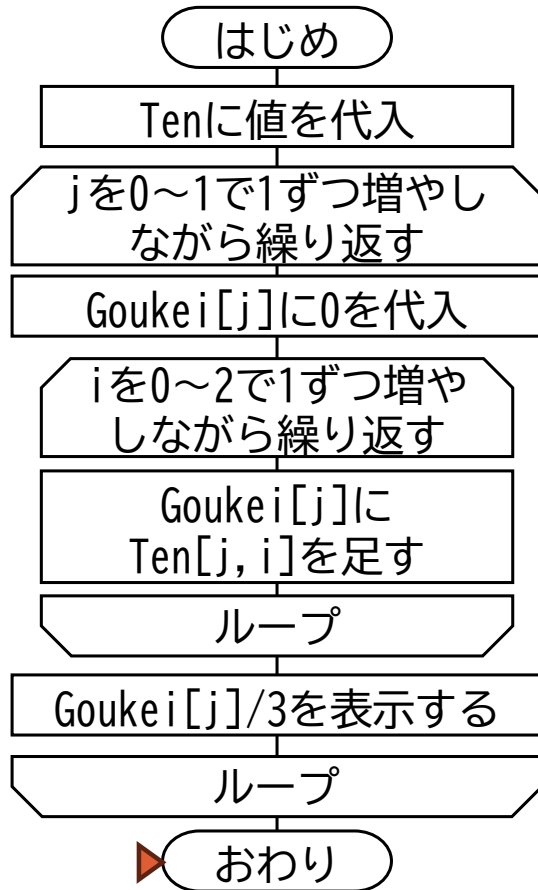
赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



変数の意味の把握、入れ子の反復処理がキモ



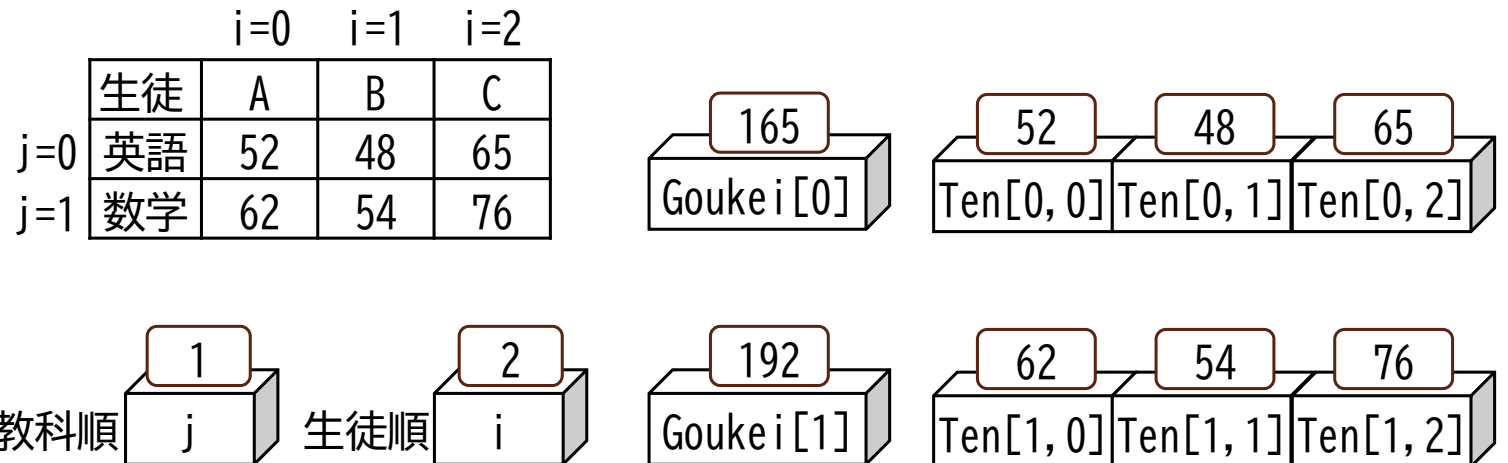
共通テストで使われるDNCL

```

Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながらか繰り返す：
  Goukei[j] = 0
  iを0から2まで1ずつ増やしながらか繰り返す：
    Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
  
```

※変数の意味を把握する

- 「=」は代入の意味 ※等しいではない
- 計算をしてから変数に代入する
 - 変数は値に置き換える

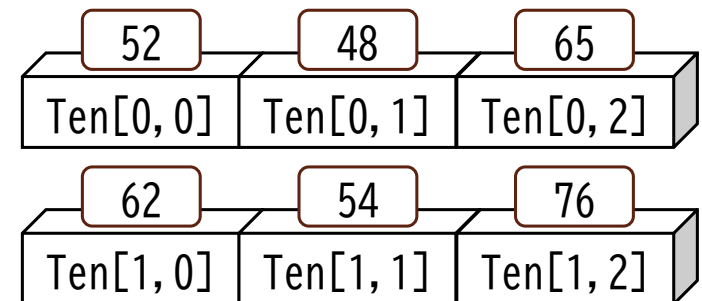
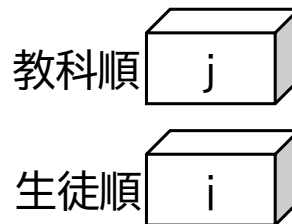


2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
```

		i=0	i=1	i=2
	生徒	A	B	C
j=0	英語	52	48	65
j=1	数学	62	54	76



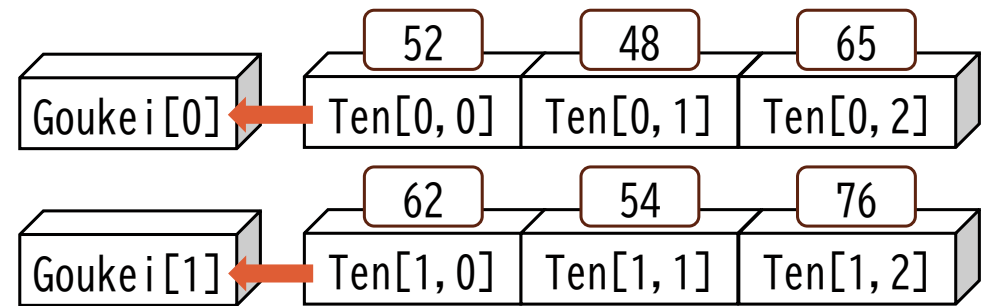
2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

教科ごとに集計していく



教科ごとに集計していく

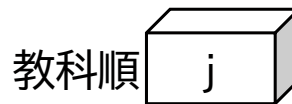
2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
```

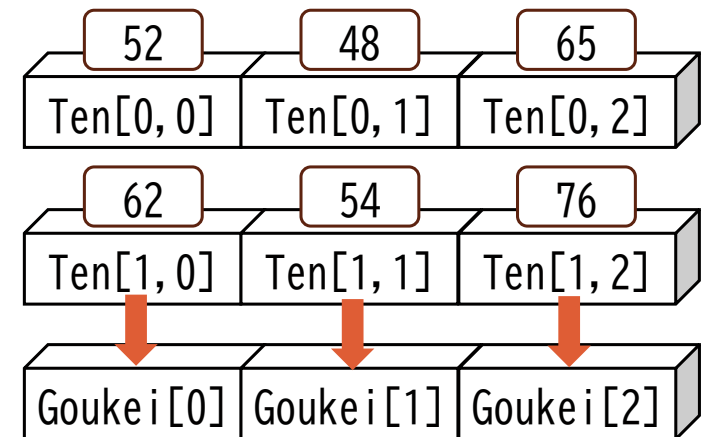
	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

生徒ごとに集計していく



生徒ごとに集計していく

生徒ごとの平均点を求めるプログラムの場合



2次元配列では添字の変数の意味を理解することが重要

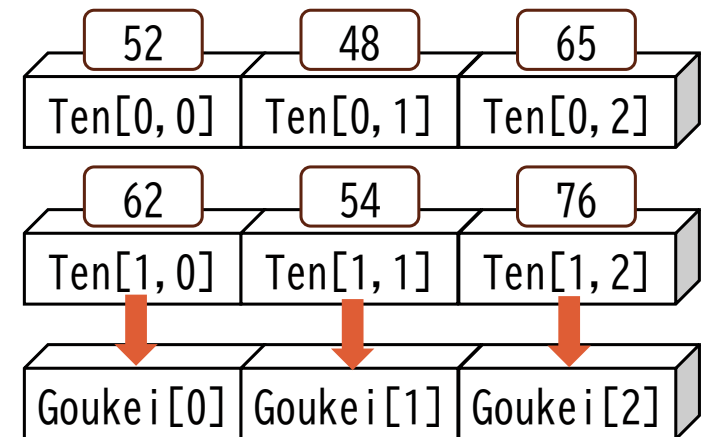
教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

小さなループ
教科順 j
生徒順 i
生徒ごとに教科を変えて集計する

生徒ごとの平均点を求めるプログラムの場合



2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

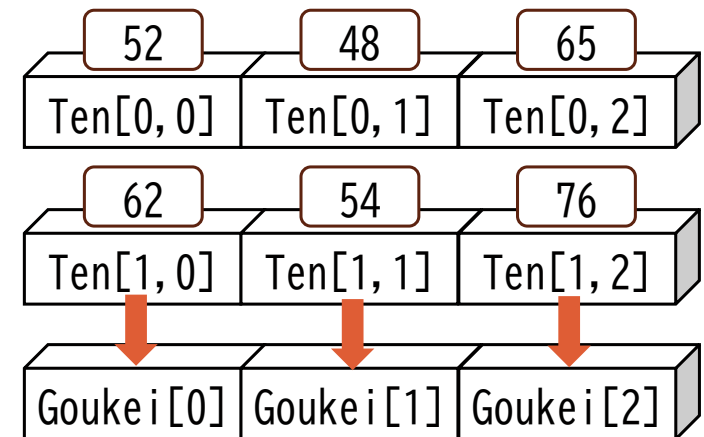
```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

小さなループ
教科順 j
生徒ごとに教科を変えて集計する

生徒順 i
大きなループ
生徒ごとに集計を繰り返す

生徒ごとの平均点を求めるプログラムの場合



2次元配列では添字の変数の意味を理解することが重要

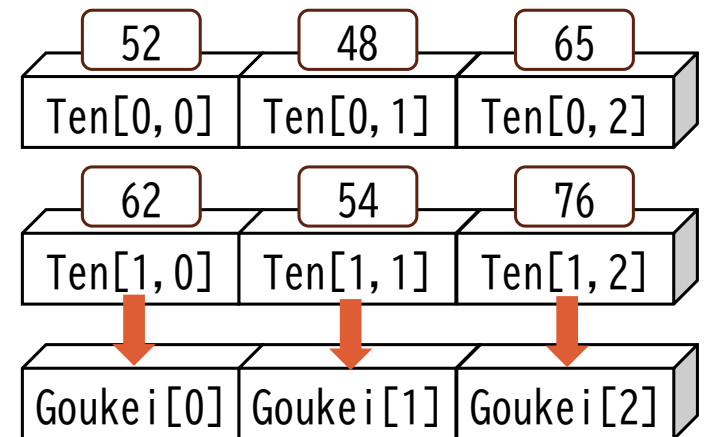
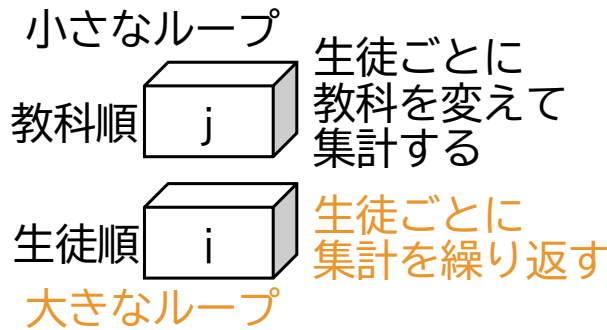
教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
│   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
│   └ 表示する(Goukei[j]/3)
```

生徒ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
iを0から2まで1ずつ増やしながら繰り返す：
┌
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76



2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
│   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
│   └ 表示する(Goukei[j]/3)
```

生徒ごとの平均点を求めるプログラム

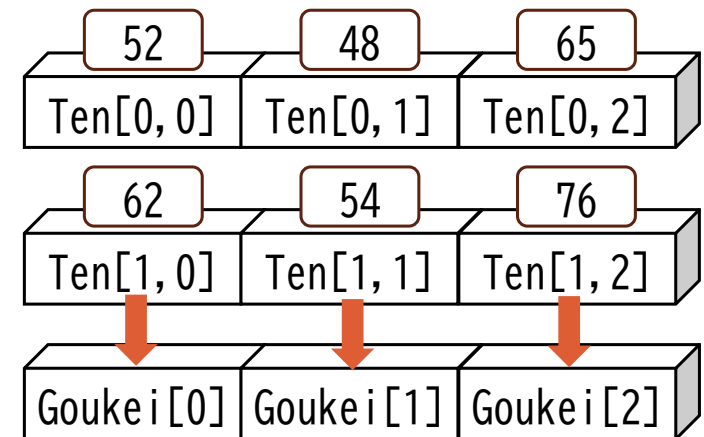
```
Ten = [ [52, 48, 65], [62, 54, 76] ]
iを0から2まで1ずつ増やしながら繰り返す：
┌ Goukei[i] = 0
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

小さなループ
教科順 j
生徒順 i
大きなループ

生徒ごとに教科を変えて集計する

生徒ごとに集計を繰り返す



2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

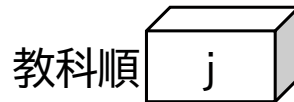
```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
└ Goukei[j] = Goukei[j] + Ten[j, i]
  表示する(Goukei[j]/3)
```

生徒ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
iを0から2まで1ずつ増やしながら繰り返す：
┌ Goukei[i] = 0
├ jを0から1まで1ずつ増やしながら繰り返す：
└
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

小さなループ

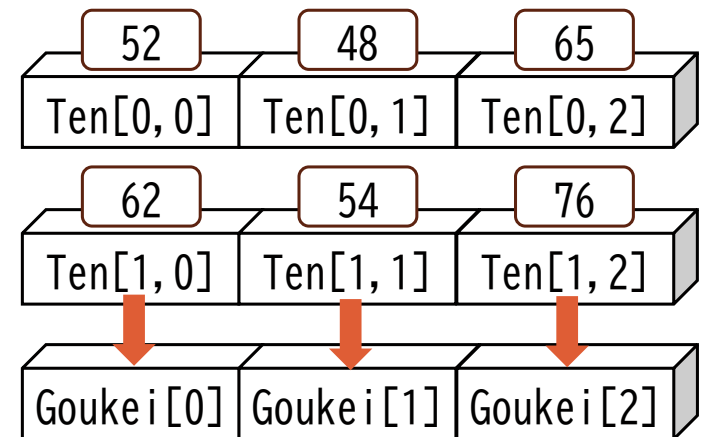


生徒ごとに教科を変えて集計する



生徒ごとに集計を繰り返す

大きなループ



2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

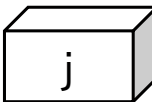
```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
├ iを0から2まで1ずつ増やしながら繰り返す：
│   ┌ Goukei[j] = Goukei[j] + Ten[j, i]
│   └ 表示する(Goukei[j]/3)
```

生徒ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
iを0から2まで1ずつ増やしながら繰り返す：
┌ Goukei[i] = 0
├ jを0から1まで1ずつ増やしながら繰り返す：
│   ┌ Goukei[i] = Goukei[i] + Ten[j, i]
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

小さなループ

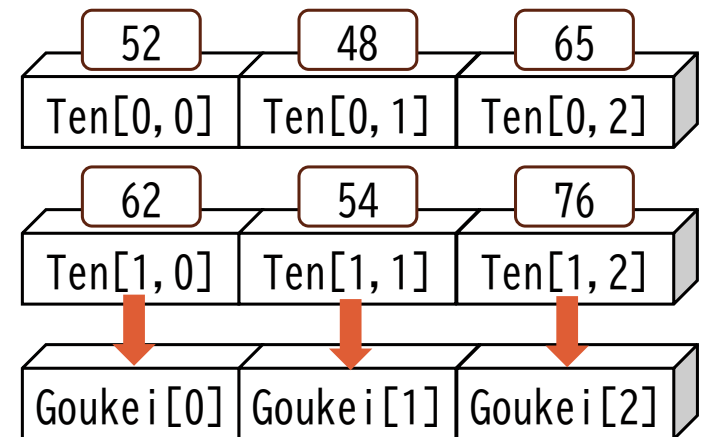
教科順 

生徒ごとに
教科を変えて
集計する

生徒順 

生徒ごとに
集計を繰り返す

大きなループ



2次元配列では添字の変数の意味を理解することが重要

教科ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
jを0から1まで1ずつ増やしながら繰り返す：
┌ Goukei[j] = 0
└ iを0から2まで1ずつ増やしながら繰り返す：
    ┌ Goukei[j] = Goukei[j] + Ten[j, i]
    └ 表示する(Goukei[j]/3)
```

生徒ごとの平均点を求めるプログラム

```
Ten = [ [52, 48, 65], [62, 54, 76] ]
iを0から2まで1ずつ増やしながら繰り返す：
┌ Goukei[i] = 0
└ jを0から1まで1ずつ増やしながら繰り返す：
    ┌ Goukei[i] = Goukei[i] + Ten[j, i]
    └ 表示する(Goukei[i]/2)
```

	i=0	i=1	i=2
生徒	A	B	C
j=0 英語	52	48	65
j=1 数学	62	54	76

小さなループ

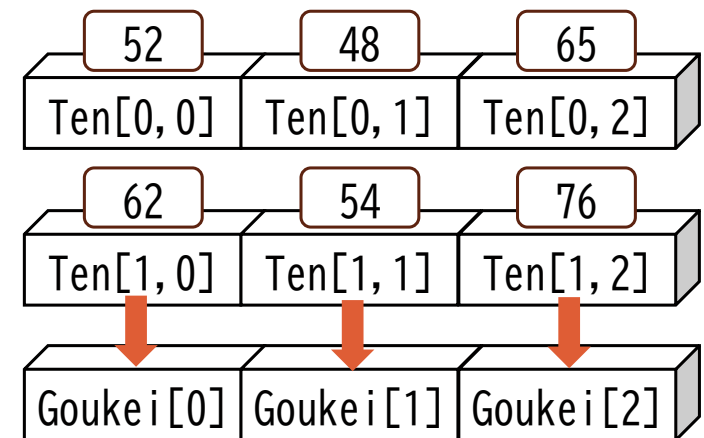


生徒ごとに教科を変えて集計する



生徒ごとに集計を繰り返す

大きなループ



「プログラミング(二次元配列)」の要点

「プログラミング(二次元配列)」の要点

生徒	A	B	C
英語	52	48	65
数学	62	54	76
国語	74	55	58

※特に添字の2つの変数の意味を把握する

添字が手前の「教科」のグループ

配列 $Ten = [[52, 48, 65], [62, 54, 76], [74, 55, 58]]$

添字が後ろの「生徒」が「教科」の中に入る

教科順 生徒順

$Ten[0,0]=52$ $Ten[0,1]=48$ $Ten[0,2]=65$

行番号 列番号

$Ten[1,0]=62$ $Ten[1,1]=54$ $Ten[1,2]=76$

$Ten[2,0]=74$ $Ten[2,1]=55$ $Ten[2,2]=58$

