速習！情報 ～共通テスト対策講座～

プログラミング（配列）

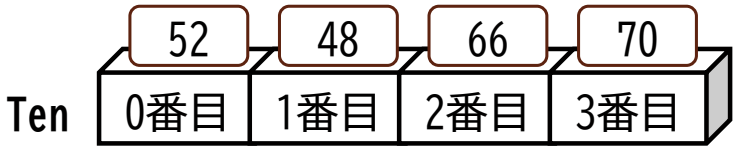
配列、配列名、添字、インデックス

配列は複数の変数をまとめたもの

点数の合計を求めたい。

生徒	山田	鈴木	田中	中村
点数	52	48	66	70

イメージ

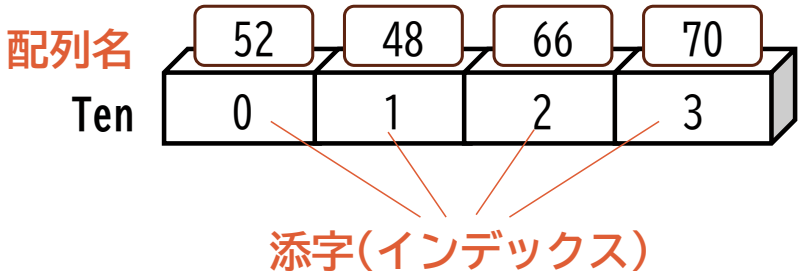


配列は複数の変数をまとめたもので、配列名[添字]で個々の値にアクセスする

点数の合計を求めたい。

生徒	山田	鈴木	田中	中村
点数	52	48	66	70

イメージ



配列　Ten=[52, 48, 66, 70]

Ten[0]=52 …… 0番目のTenの値
Ten[1]=48 …… 1番目のTenの値
Ten[2]=66 …… 2番目のTenの値
Ten[3]=70 …… 3番目のTenの値

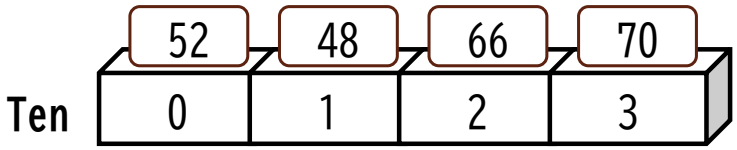
- 添字の開始が0のプログラミング言語と1のものがある
- 共通テストのDNCLでは0開始が標準だが、1開始の問題もある

配列は複数の変数をまとめたもので、配列名[添字]で個々の値にアクセスする

点数の合計を求めたい。

生徒	山田	鈴木	田中	中村
点数	52	48	66	70

イメージ

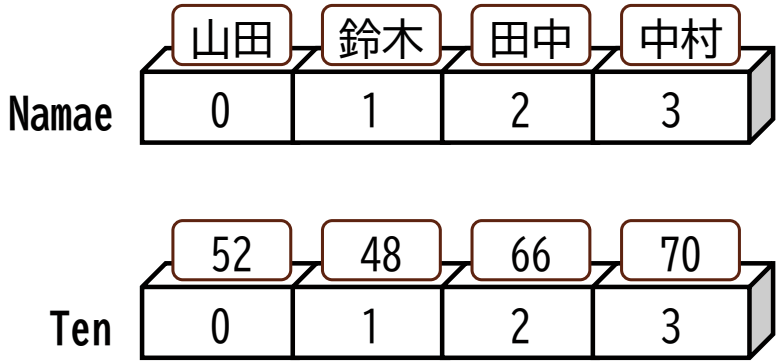


配列は複数の変数をまとめたもので、配列名[添字]で個々の値にアクセスする

点数の合計を求めたい。

生徒	山田	鈴木	田中	中村
点数	52	48	66	70

イメージ

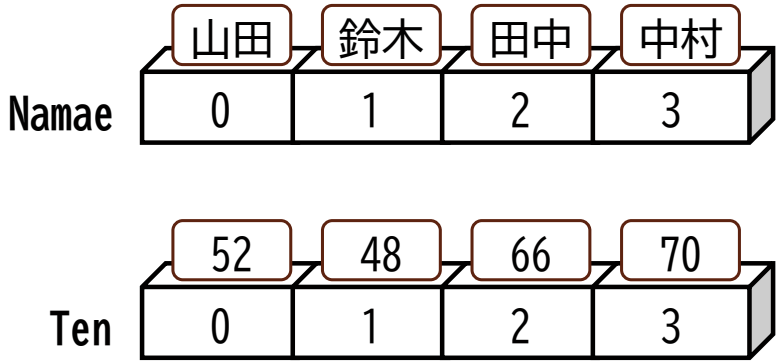


配列は複数の変数をまとめたもので、配列名[添字]で個々の値にアクセスする

点数の合計を求めたい。

生徒	山田	鈴木	田中	中村
点数	52	48	66	70

イメージ



配列　Namae=[”山田”,”鈴木”,”田中”,”中村”]

Namae[0]=”山田”
Namae[1]=”鈴木”
Namae[2]=”田中”
Namae[3]=”中村”

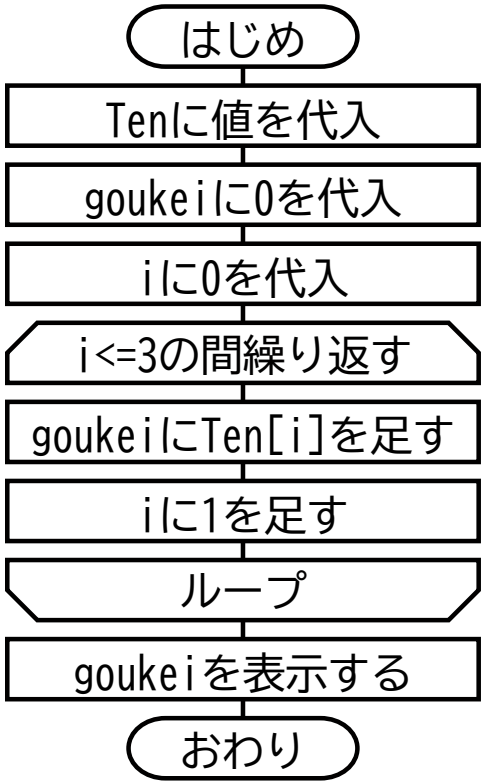
「”」で囲われたら文字列そのまま

配列　Ten=[52, 48, 66, 70]

Ten[0]=52
Ten[1]=48
Ten[2]=66
Ten[3]=70

配列は反復処理と組み合わせる

4科目の点数(52,48,66,70)の合計を求める



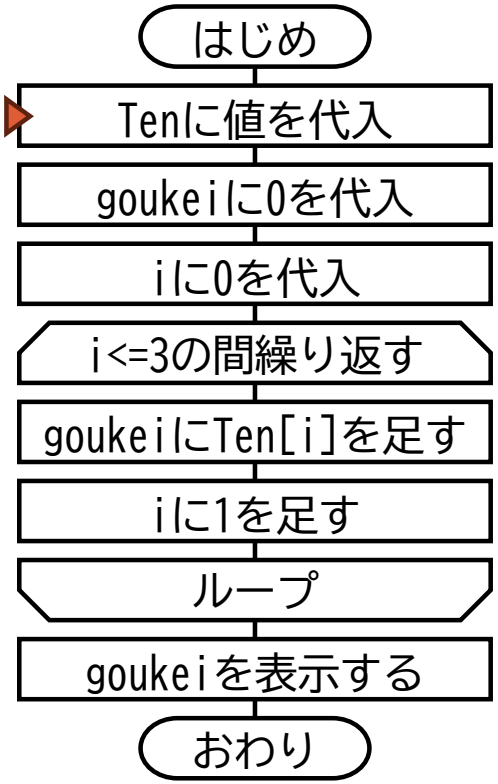
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
  goukei = goukei + Ten[i]
  i = i + 1
表示する(goukei)
```

配列は反復処理と組み合わせる

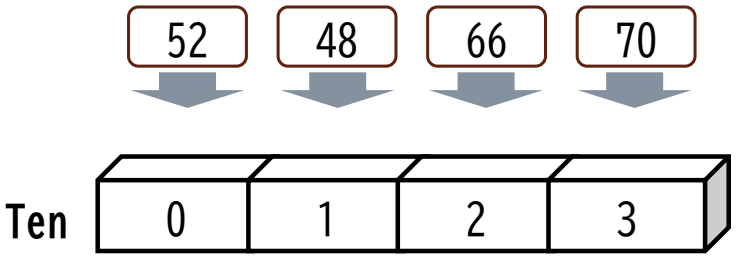
- 「＝」は代入の意味　※等しいではない

4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

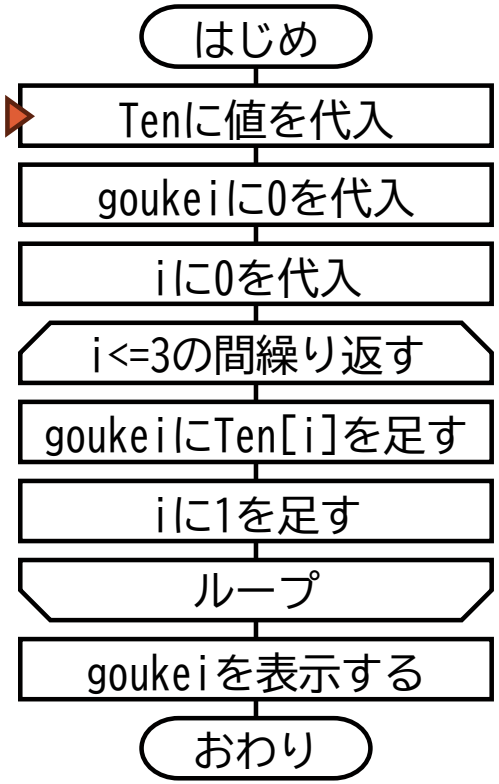
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```



配列は反復処理と組み合わせる

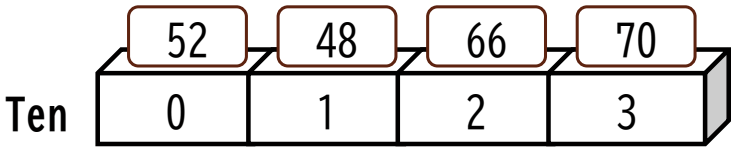
- 「＝」は代入の意味　※等しいではない

4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

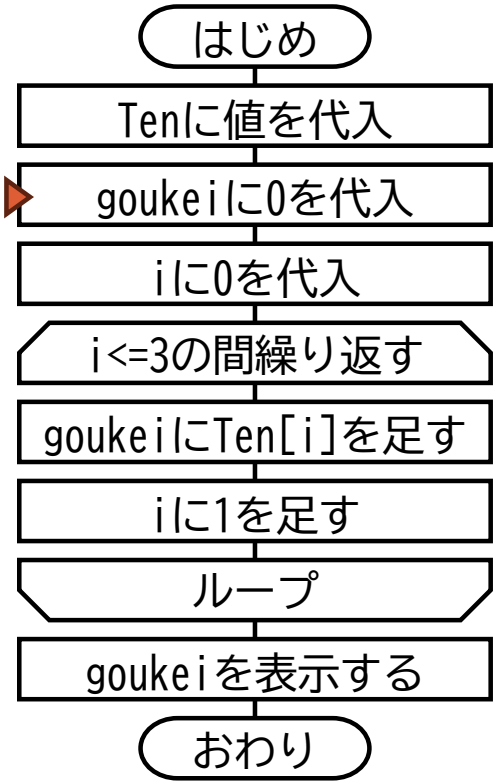
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```



配列は反復処理と組み合わせる

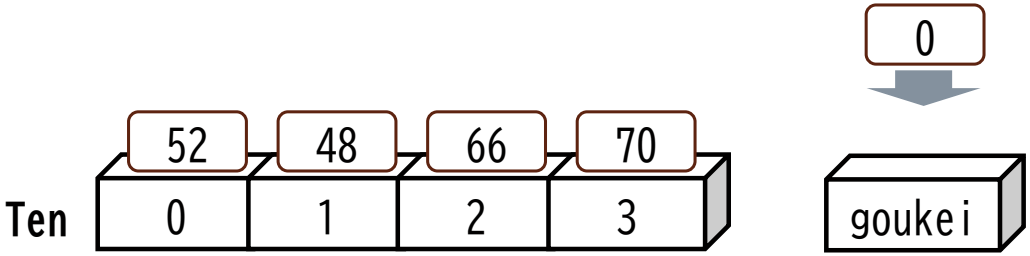
- 「＝」は代入の意味　※等しいではない

4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

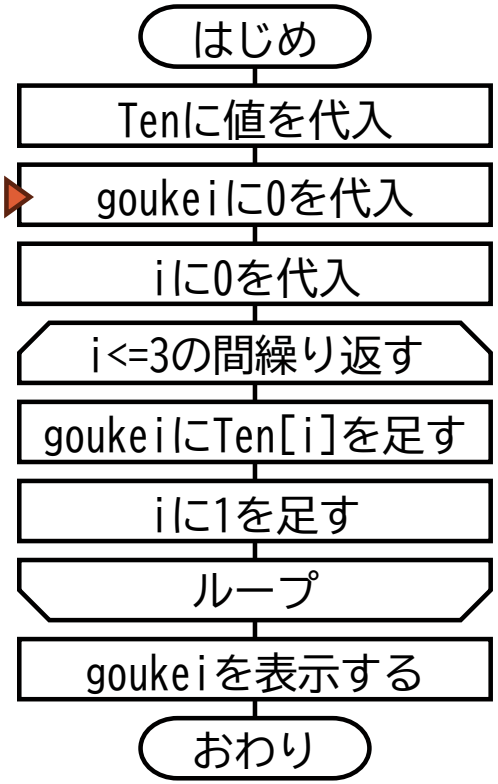
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```



配列は反復処理と組み合わせる

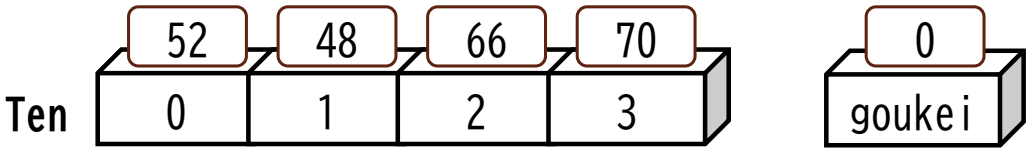
- 「＝」は代入の意味　※等しいではない

4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

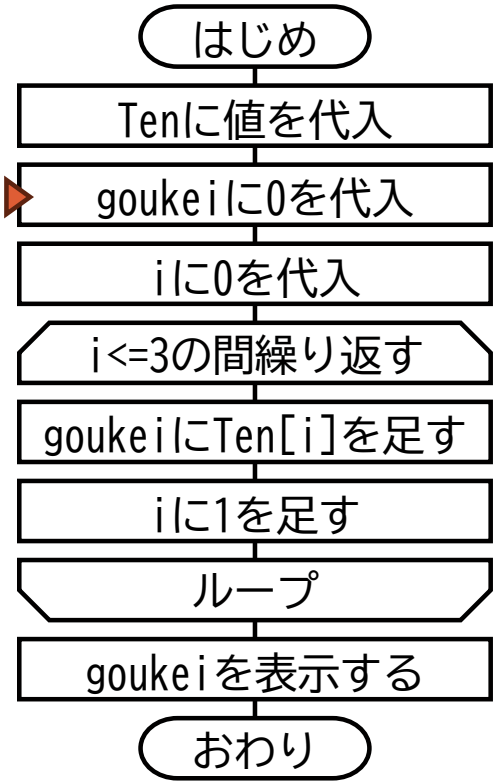
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```



配列は反復処理と組み合わせる

- 「＝」は代入の意味　※等しいではない

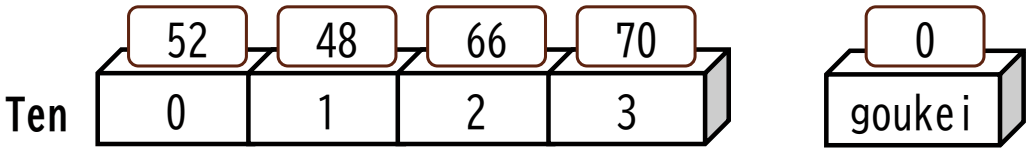
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

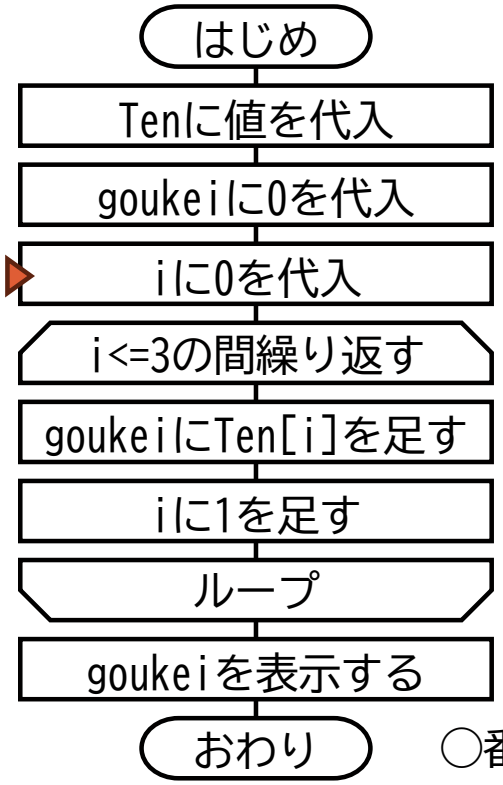
数学での計算
合計＝0



配列は反復処理と組み合わせる

- 「＝」は代入の意味　※等しいではない

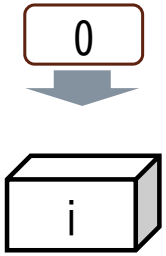
4科目の点数(52,48,66,70)の合計を求める

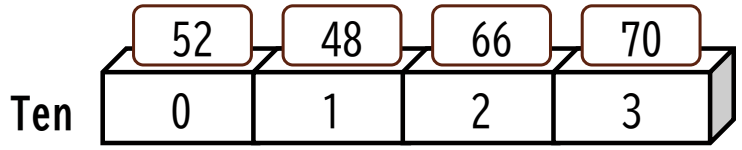


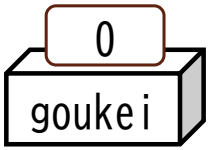
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝0



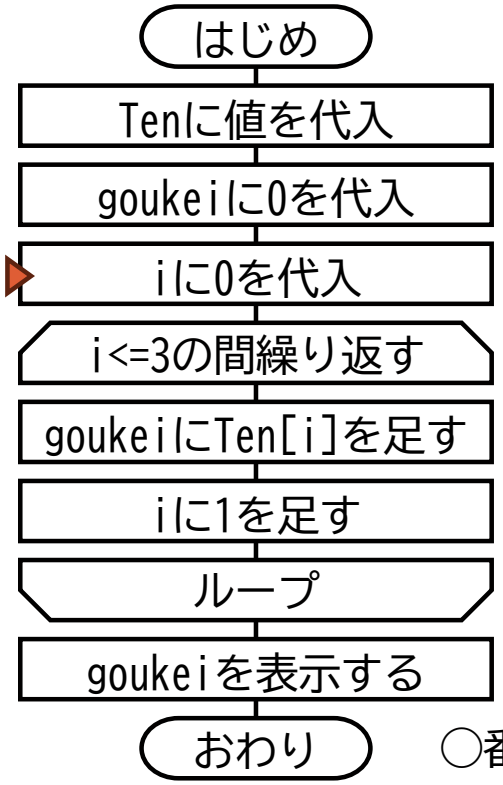




配列は反復処理と組み合わせる

- 「＝」は代入の意味　※等しいではない

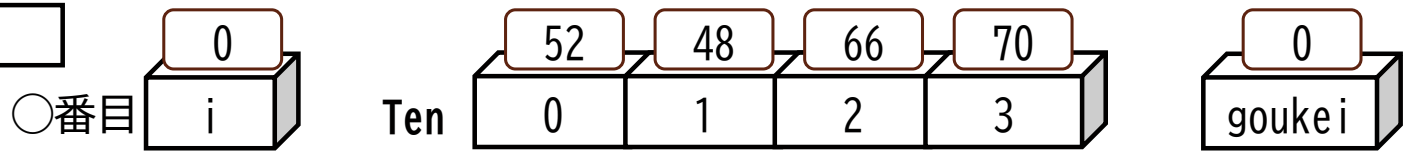
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

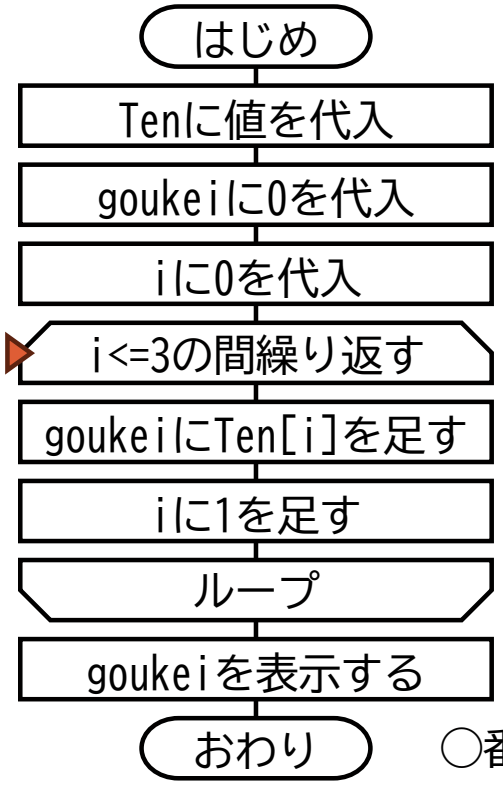
数学での計算
合計＝0



配列は反復処理と組み合わせる

- 「＝」は代入の意味　※等しいではない

4科目の点数(52,48,66,70)の合計を求める

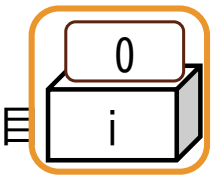


共通テストで使用されるDNCL

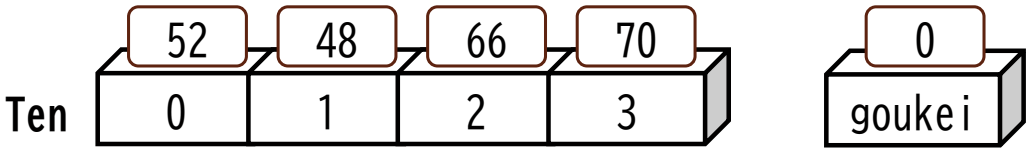
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
  goukei = goukei + Ten[i]
  i = i + 1
表示する(goukei)
```

いまのiは0で、条件を満たす

数学での計算
合計＝0



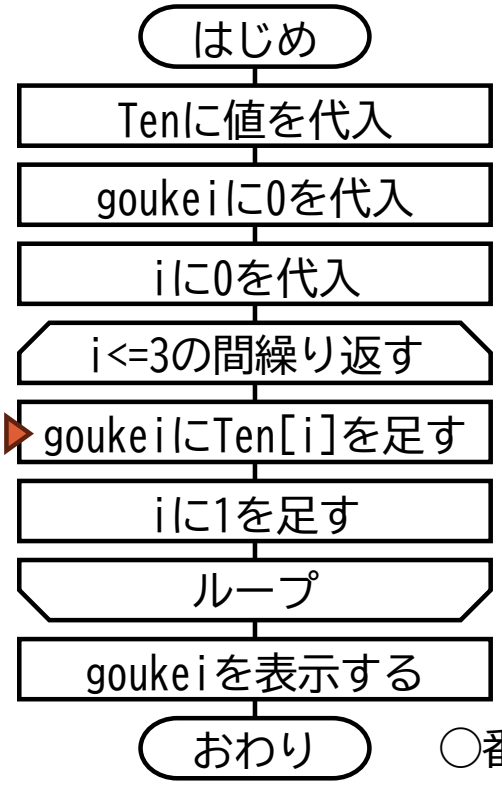
○番目



配列は反復処理と組み合わせる

- 「＝」は代入の意味　※等しいではない

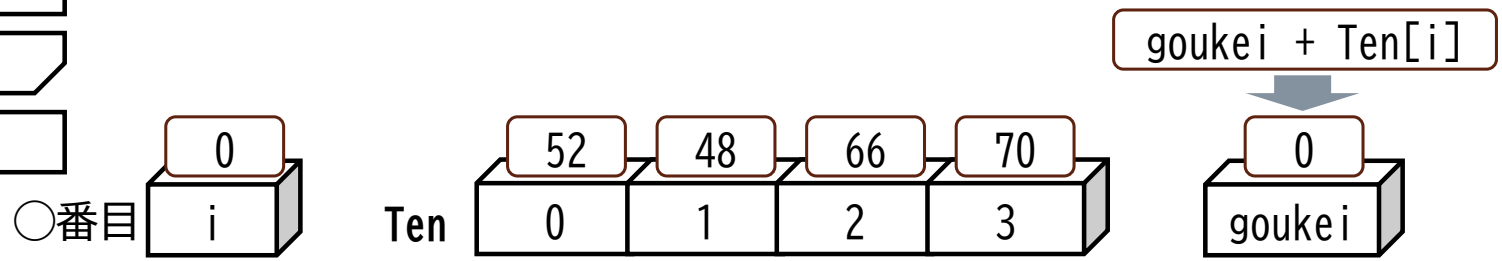
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

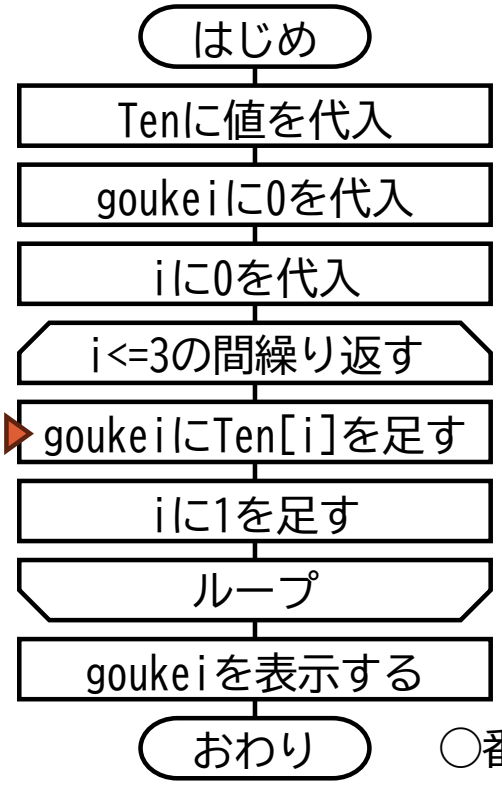
数学での計算
合計＝0



配列は反復処理と組み合わせる

- 「＝」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

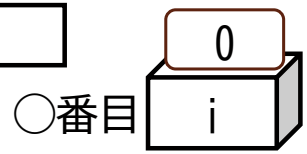
4科目の点数(52, 48, 66, 70)の合計を求める

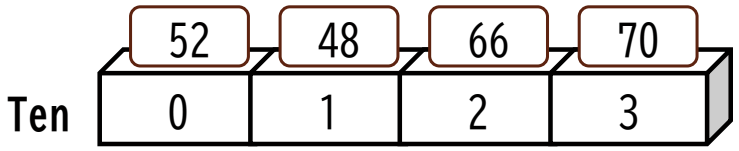


共通テストで使用されるDNCL

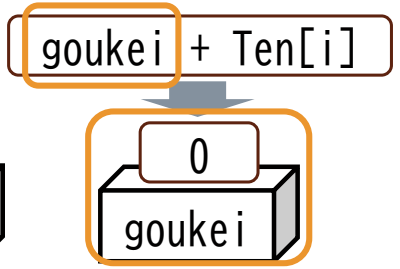
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝0





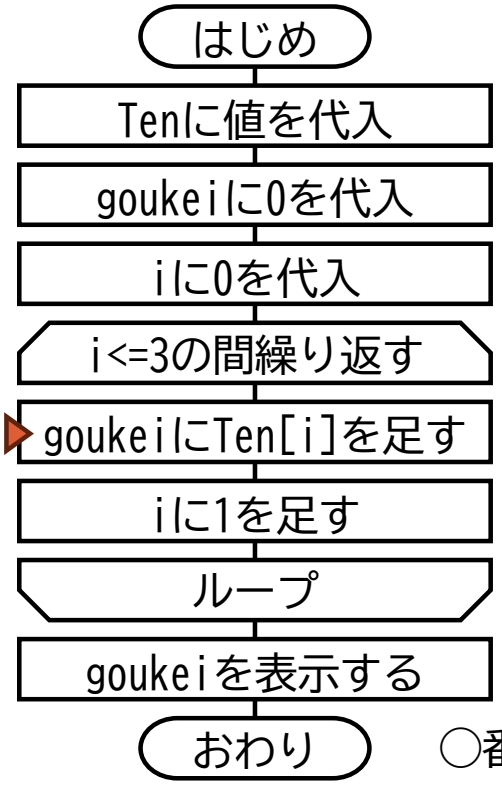
いまのgoukeiは0



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52, 48, 66, 70)の合計を求める

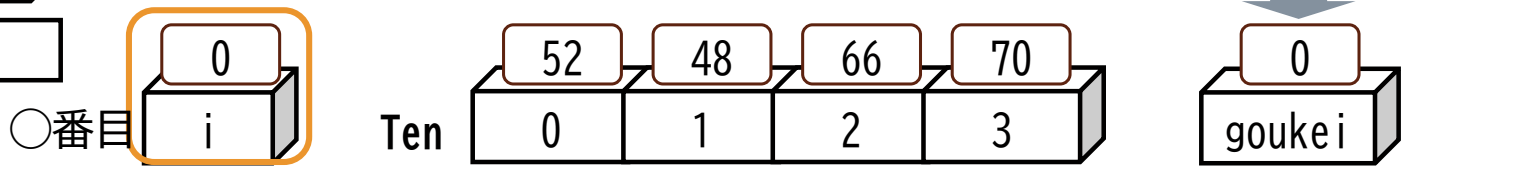


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝0

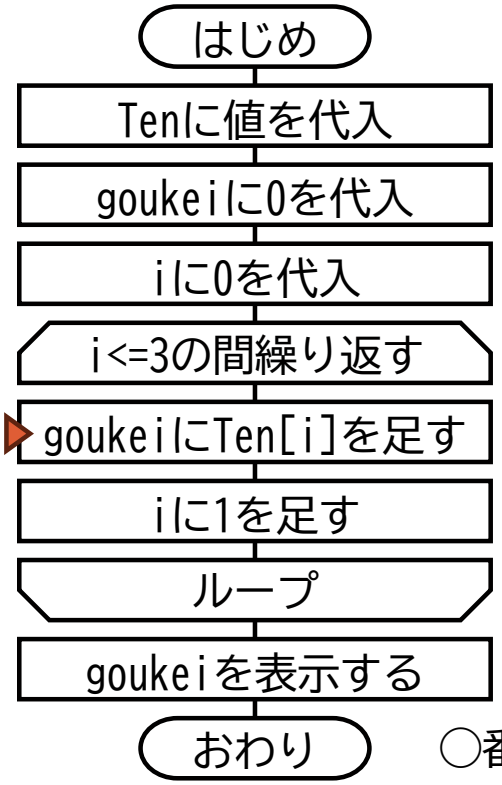
いまのiは0



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52, 48, 66, 70)の合計を求める

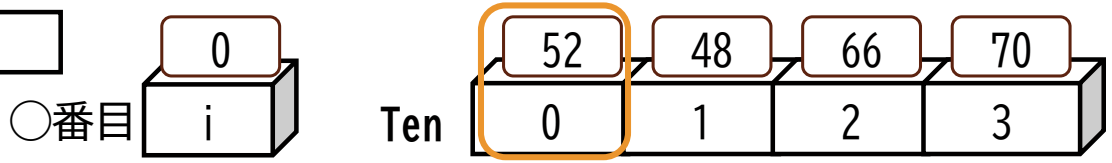


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝0

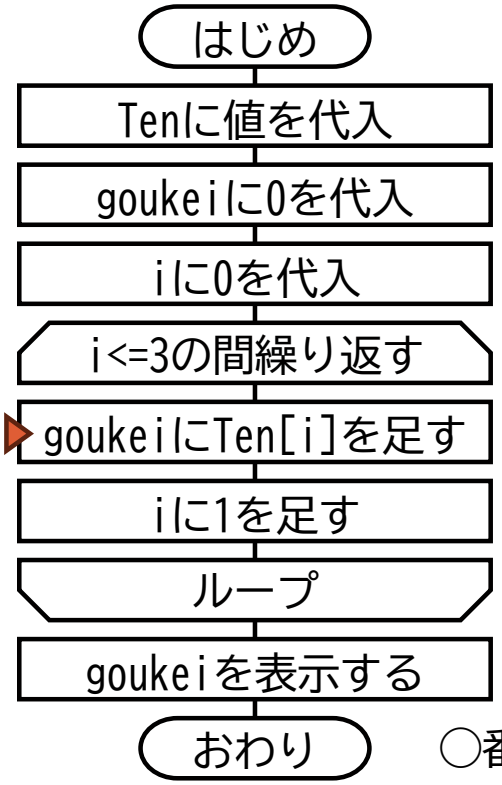
いまのTen[0]は52



配列は反復処理と組み合わせる

- 「＝」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

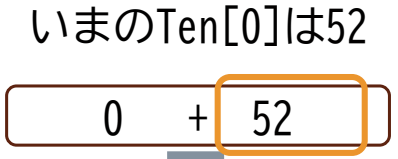
4科目の点数(52, 48, 66, 70)の合計を求める

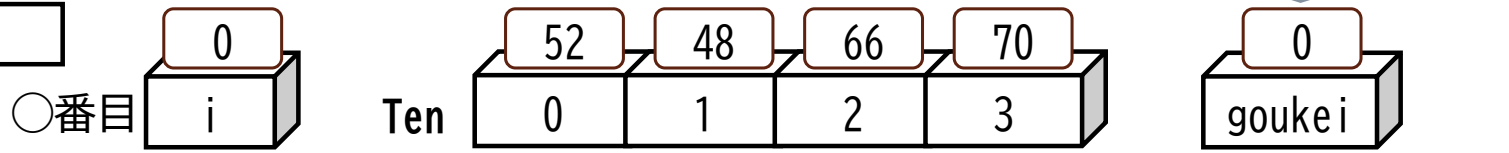


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝0

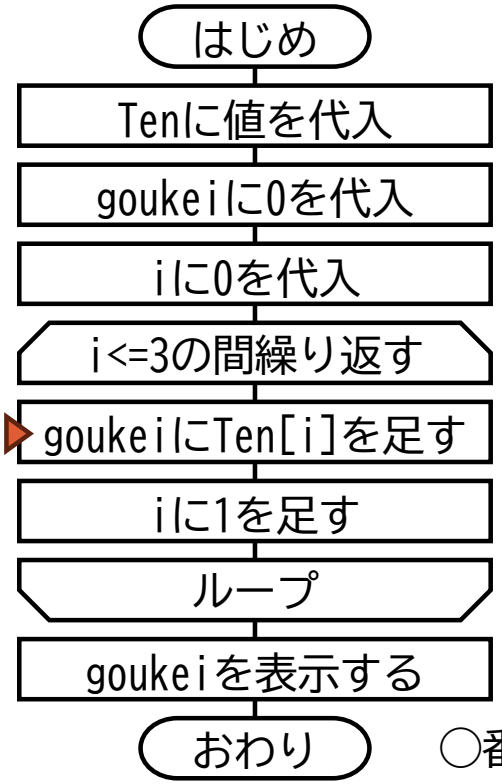




配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

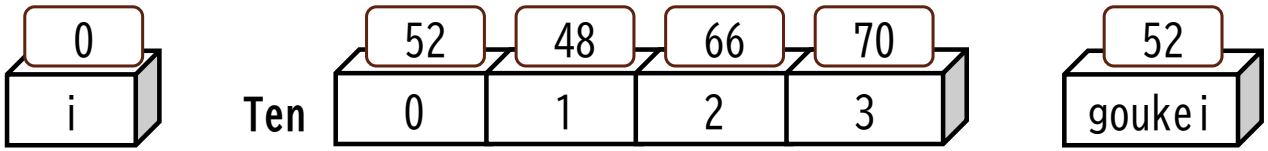


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算

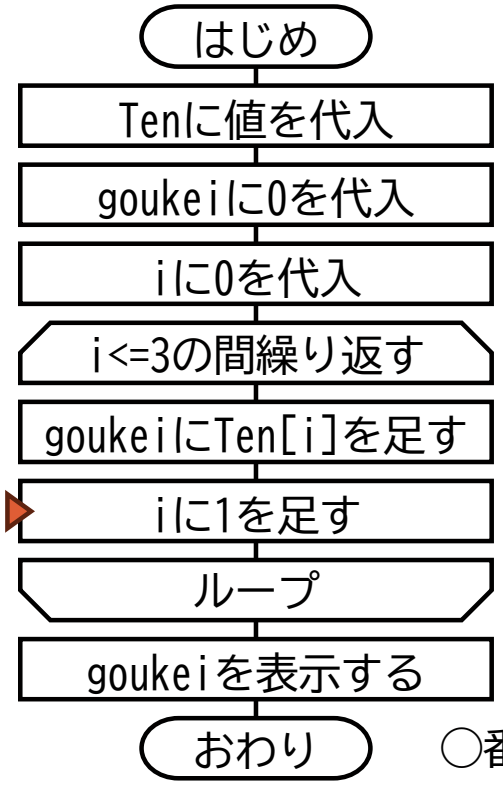
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

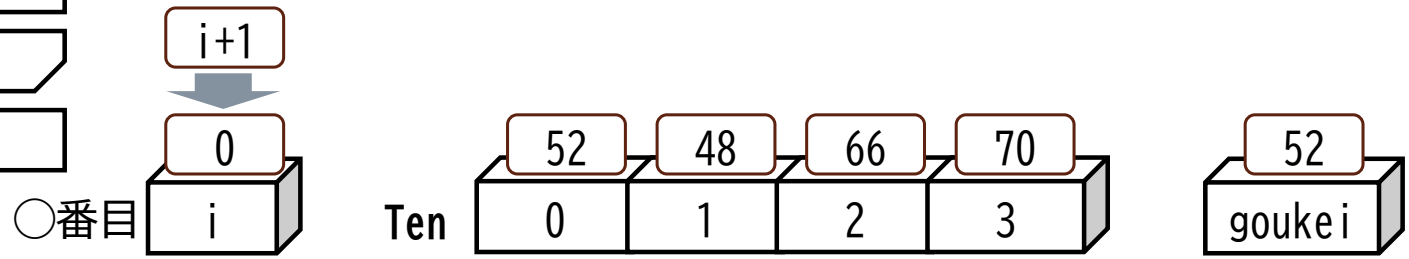
4科目の点数(52, 48, 66, 70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

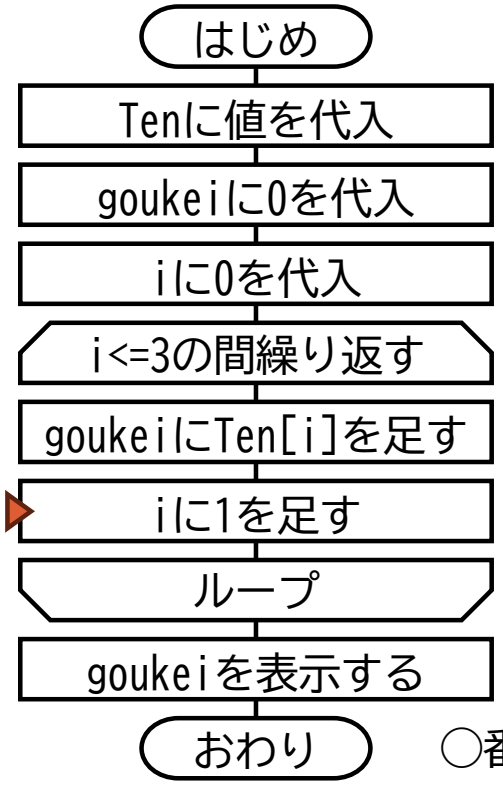
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

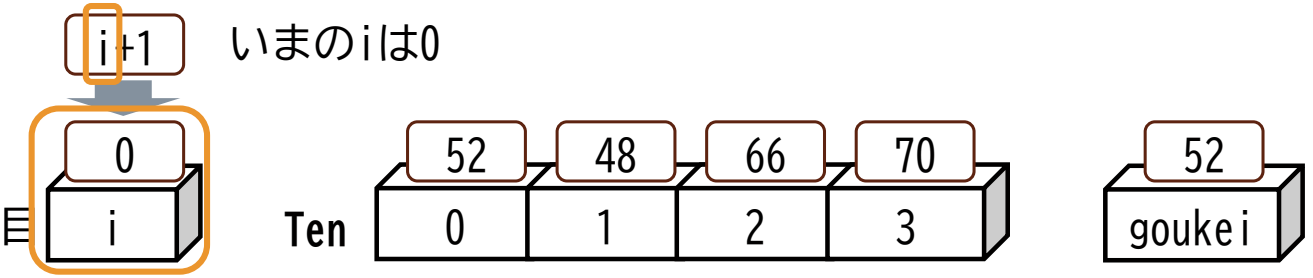
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

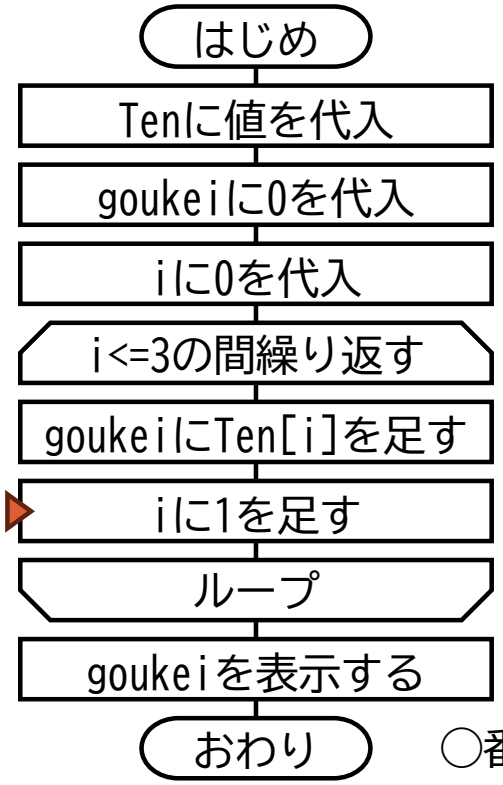
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

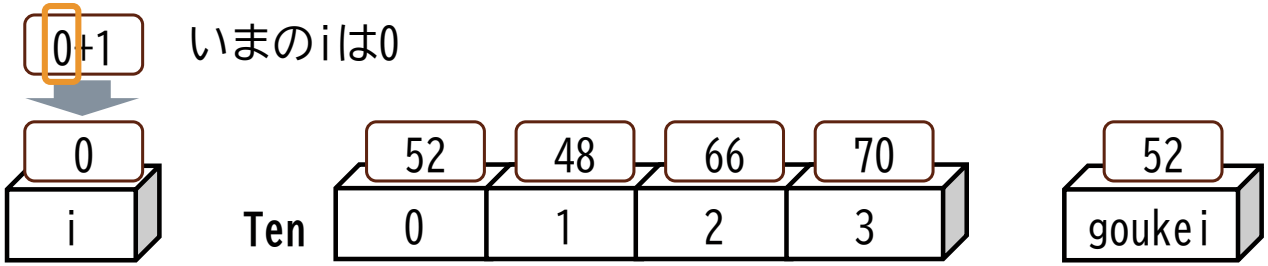
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

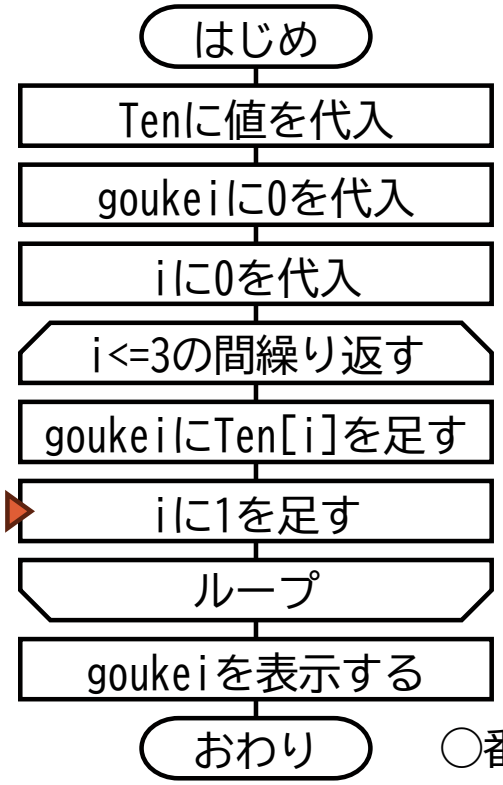
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

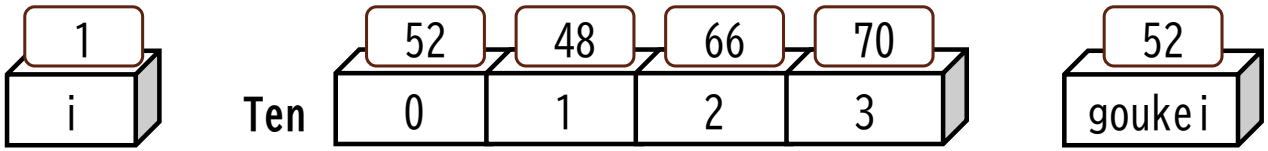
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

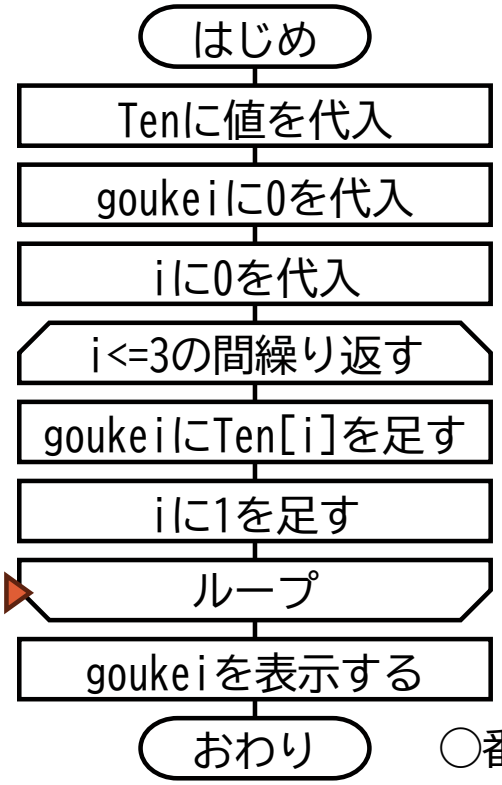
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

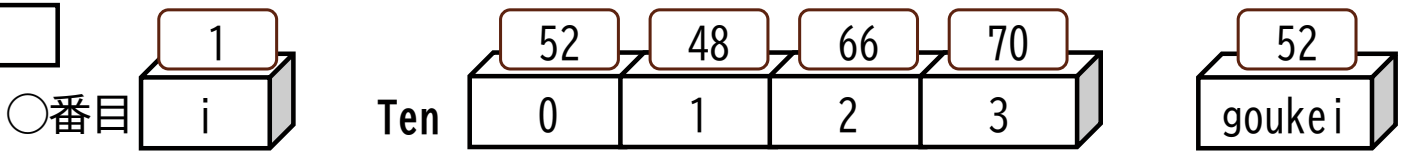


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

反復処理の終了部分を意味する

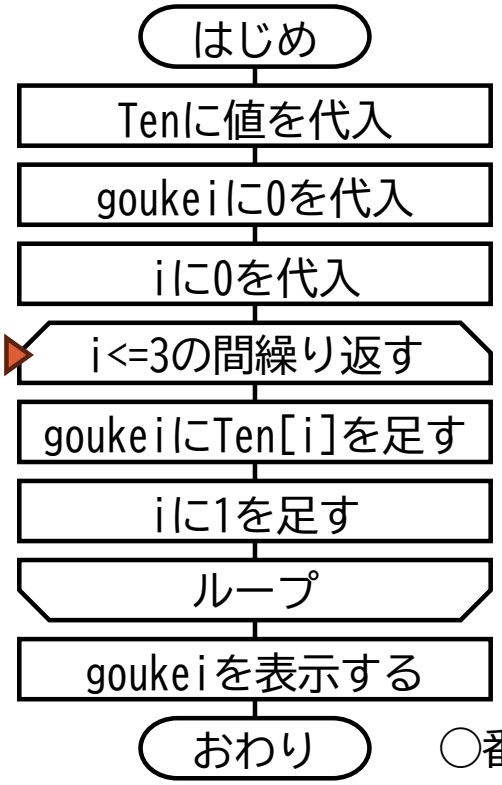
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

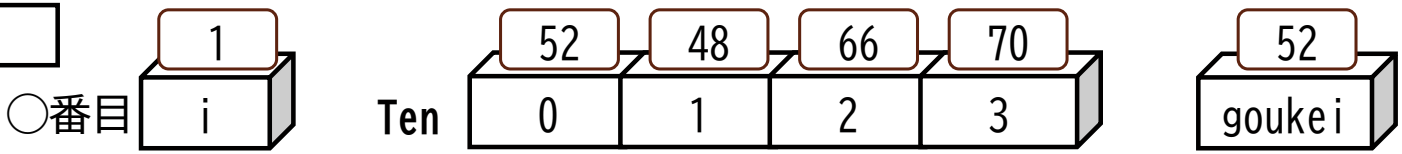
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

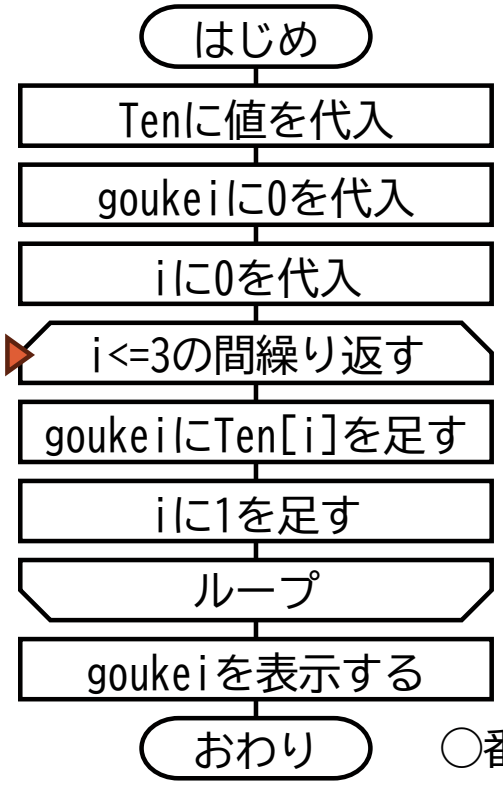
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

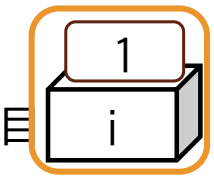


共通テストで使用されるDNCL

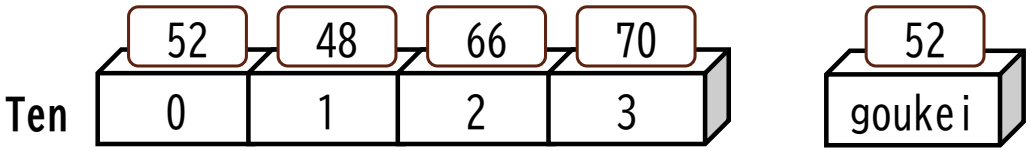
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

いまのiは1で、条件を満たす

数学での計算
合計＝52



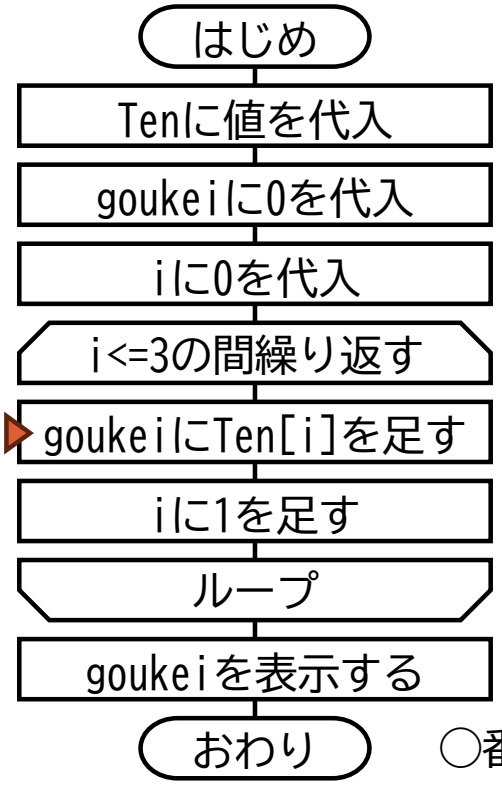
○番目



配列は反復処理と組み合わせる

- 「＝」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

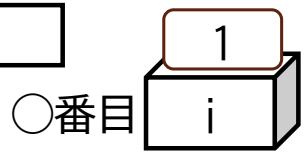
4科目の点数(52, 48, 66, 70)の合計を求める



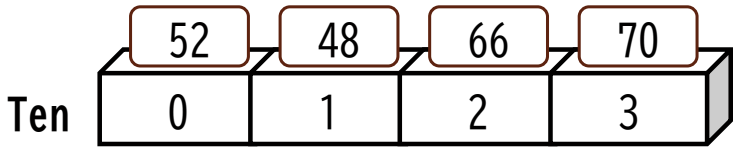
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

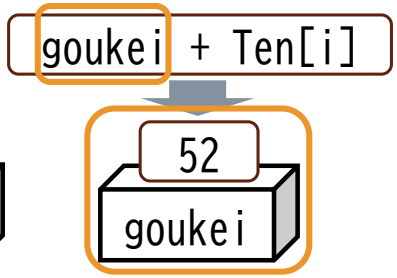
数学での計算
合計＝52



○番目



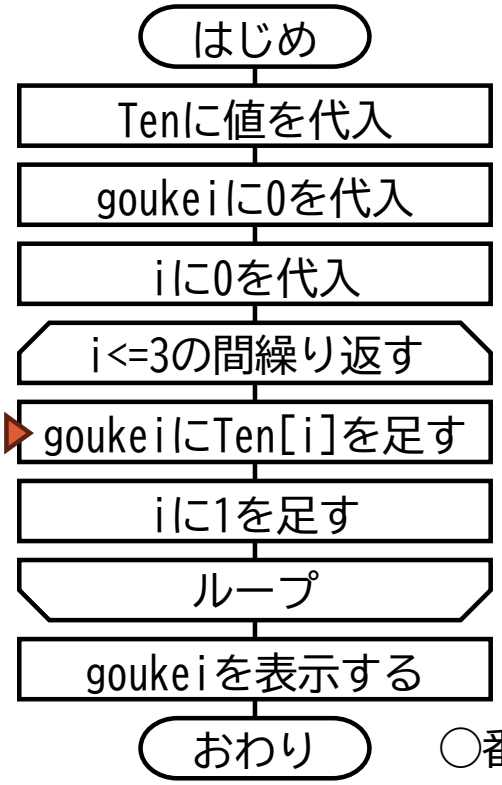
いまのgoukeiは52



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52, 48, 66, 70)の合計を求める

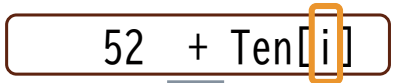


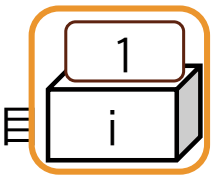
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

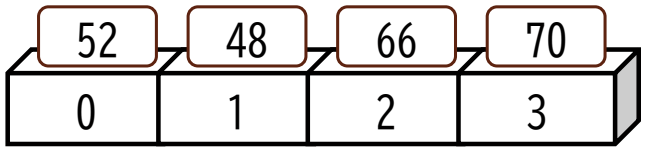
数学での計算
合計＝52

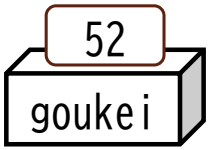
いまのiは1





○番目

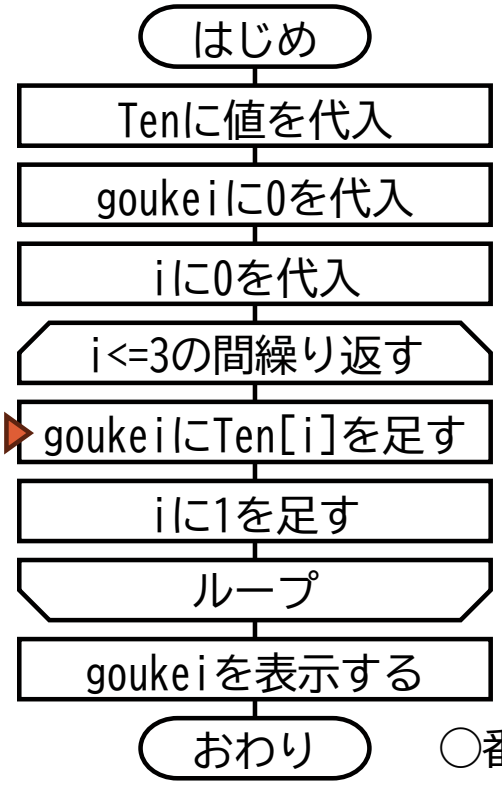




配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52, 48, 66, 70)の合計を求める

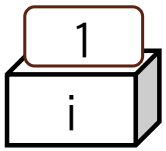


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

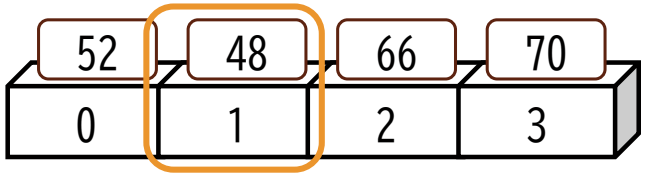
数学での計算
合計＝52

いまのTen[1]は48



○番目

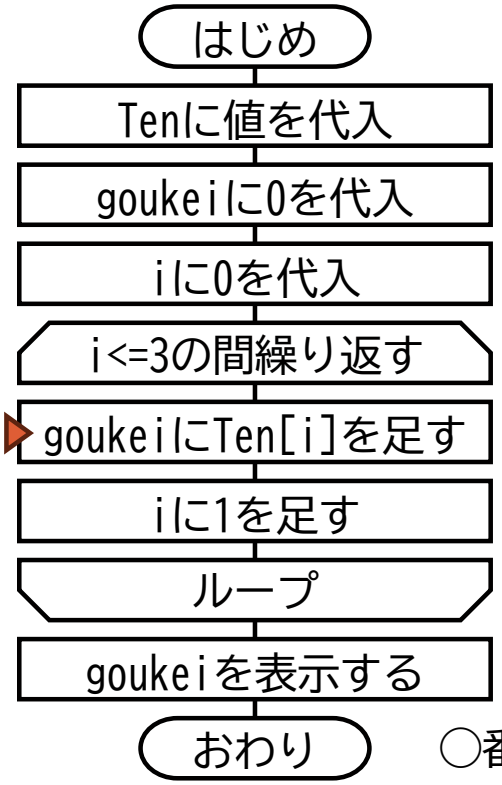
Ten



配列は反復処理と組み合わせる

- 「＝」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

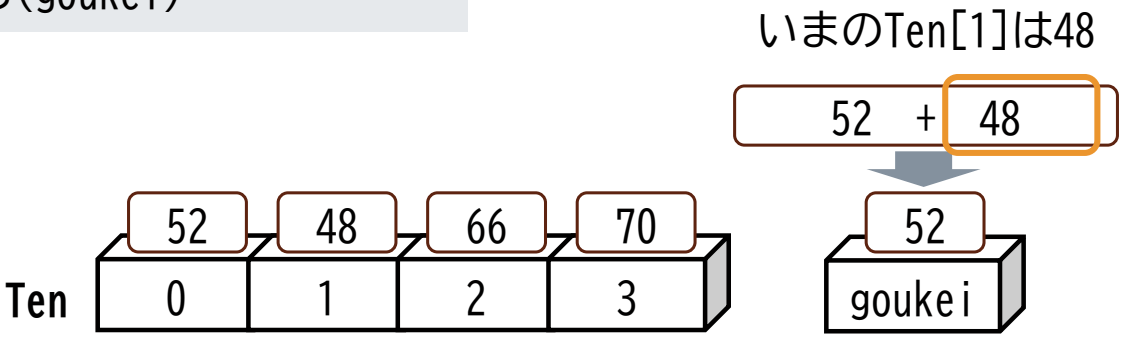
4科目の点数(52, 48, 66, 70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

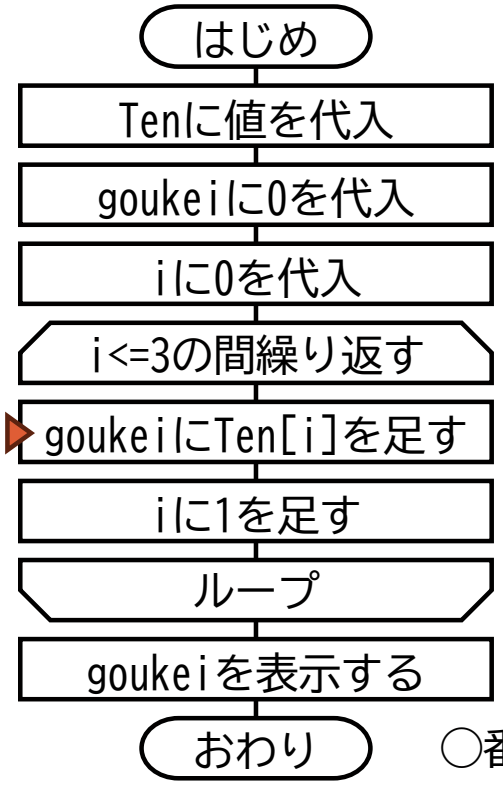
数学での計算
合計＝52



配列は反復処理と組み合わせる

- 「＝」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

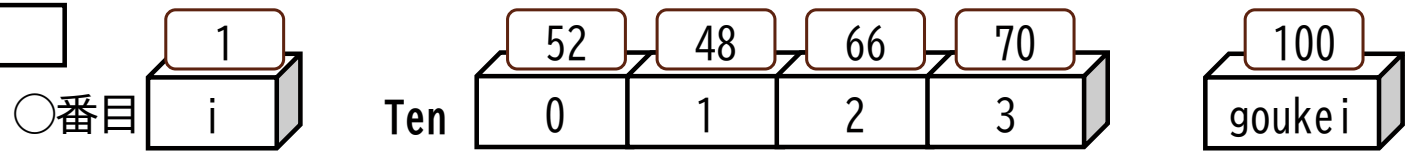
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

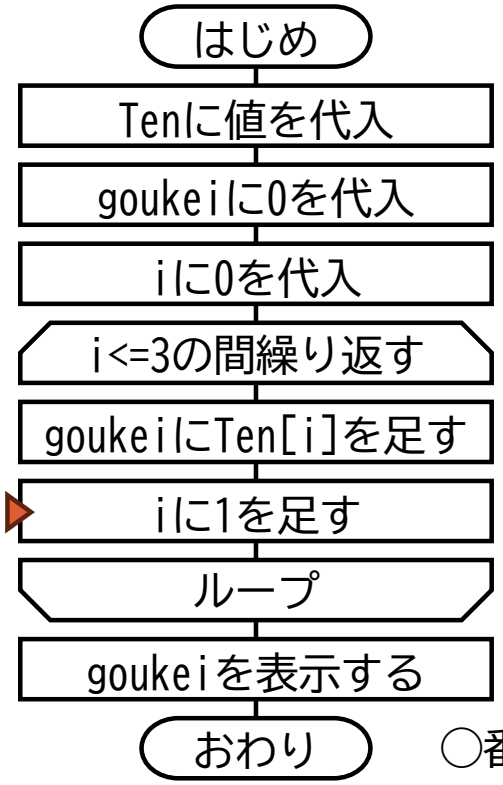
数学での計算
合計＝52＋48



配列は反復処理と組み合わせる

- 「＝」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

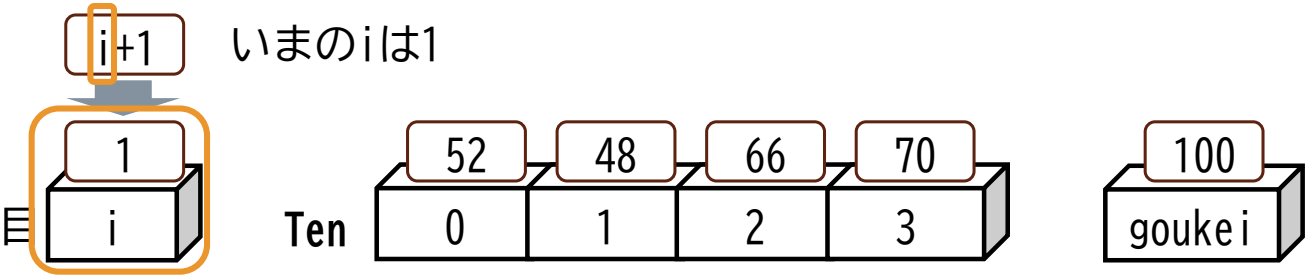
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

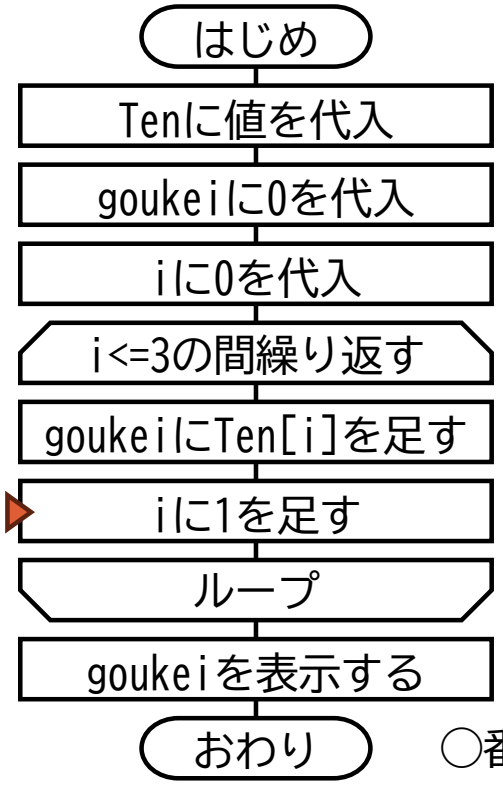
数学での計算
合計＝52＋48



配列は反復処理と組み合わせる

- 「＝」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

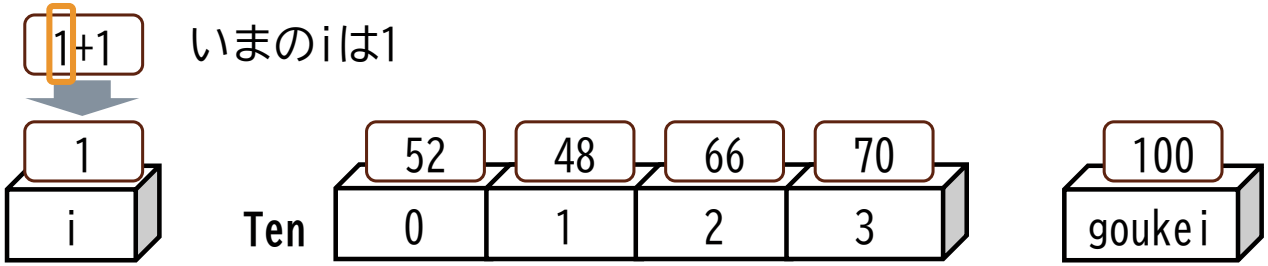
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

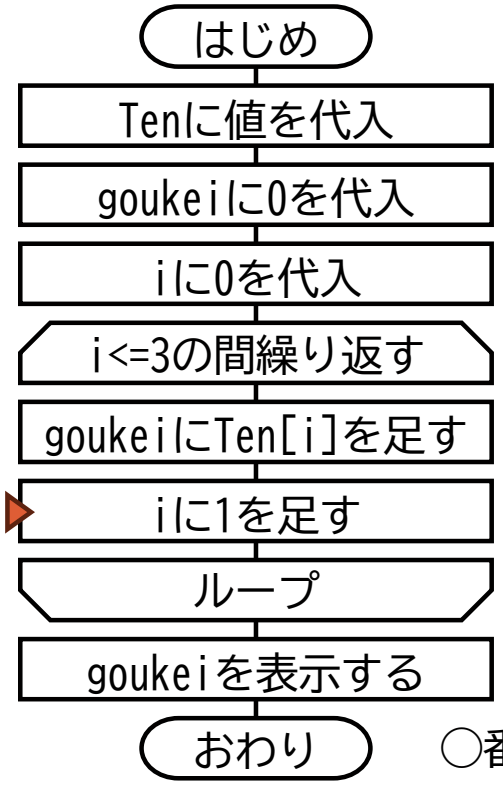
数学での計算
合計＝52＋48



配列は反復処理と組み合わせる

- 「＝」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

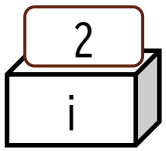
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

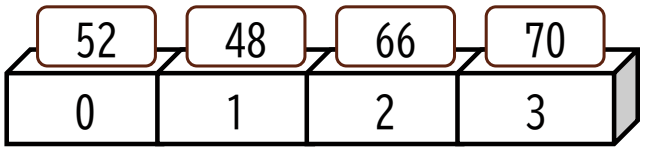
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

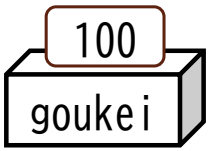
数学での計算
合計＝52＋48



○番目

Ten

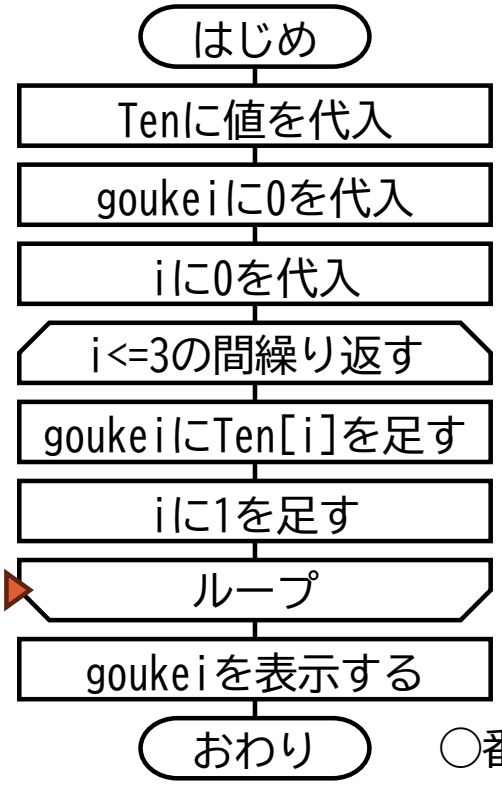




配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

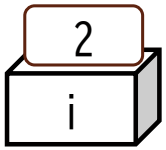


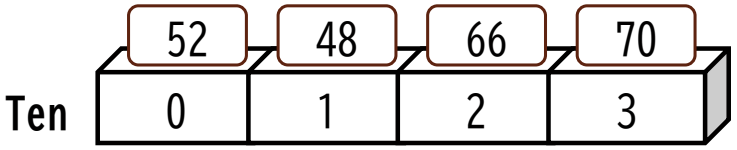
共通テストで使用されるDNCL

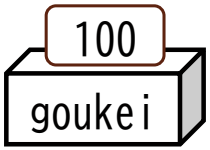
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

反復処理の終了部分を意味する

数学での計算
合計＝52＋48



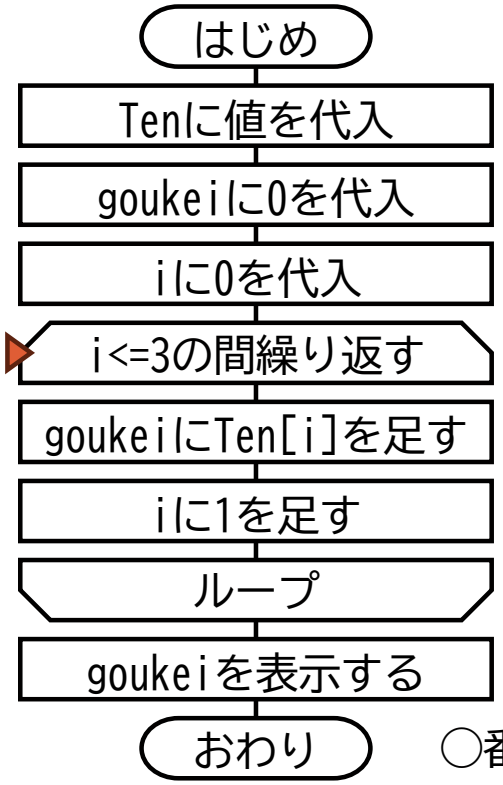




配列は反復処理と組み合わせる

- 「＝」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

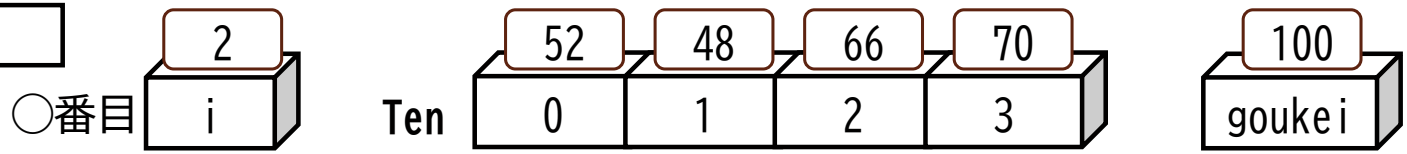
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

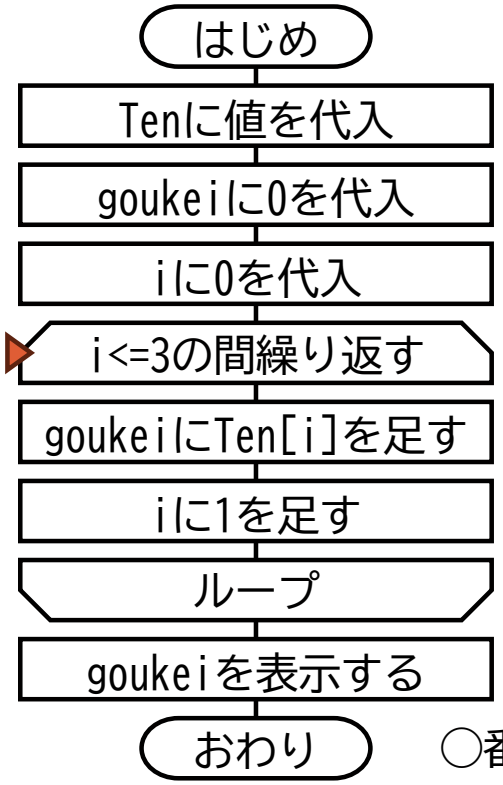
数学での計算
合計＝52＋48



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

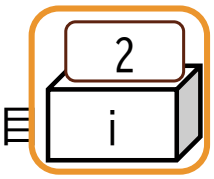


共通テストで使用されるDNCL

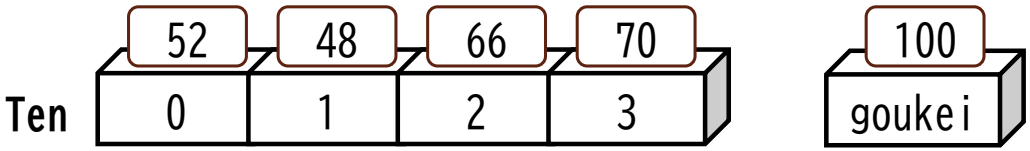
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48

いまのiは2で、条件を満たす



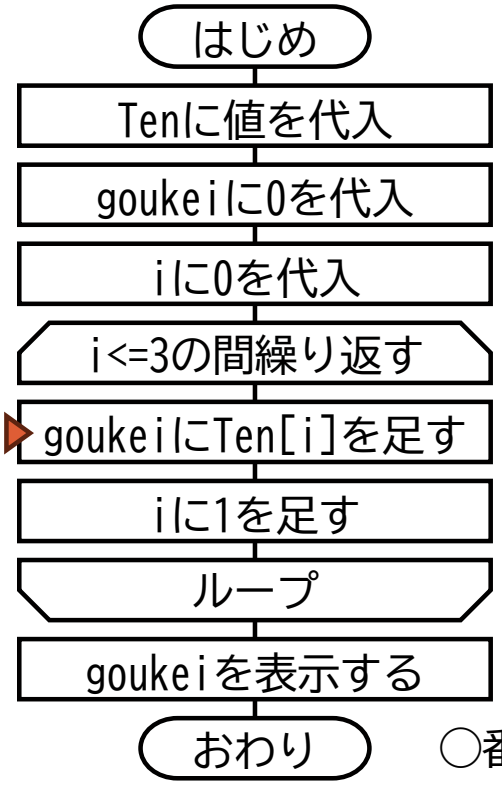
○番目



配列は反復処理と組み合わせる

- 「＝」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

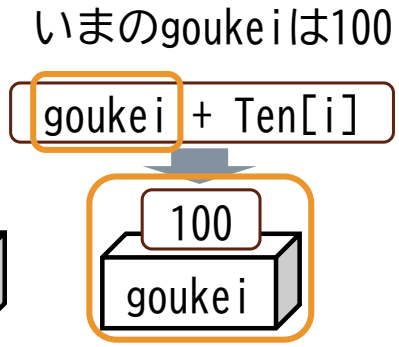
4科目の点数(52, 48, 66, 70)の合計を求める

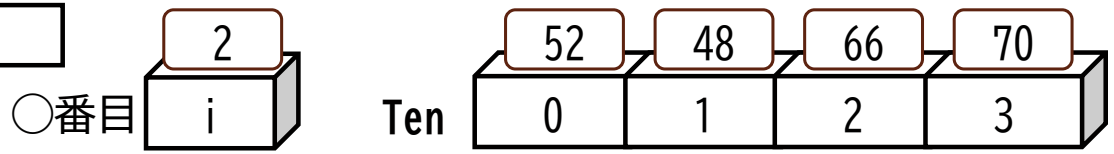


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48

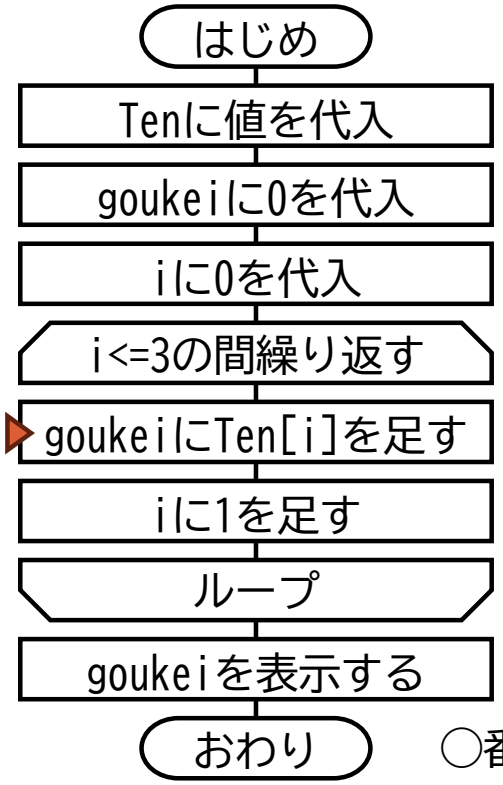




配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

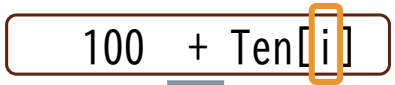


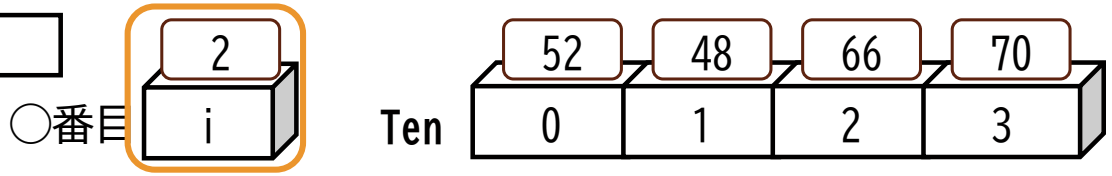
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48

いまのiは2

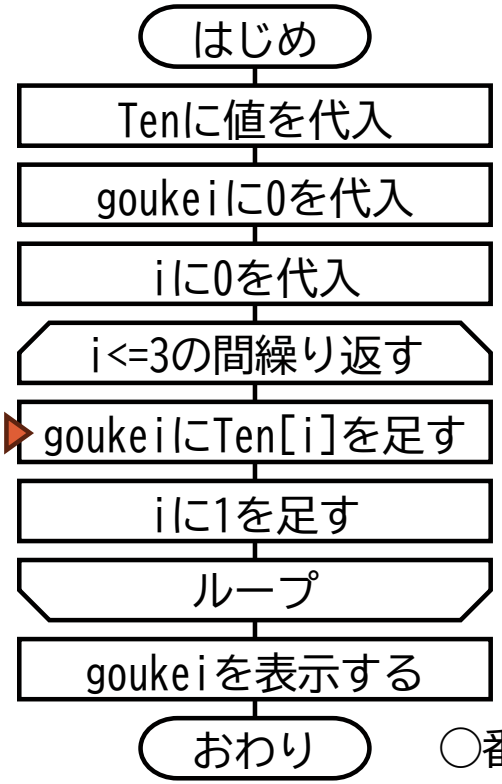




配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

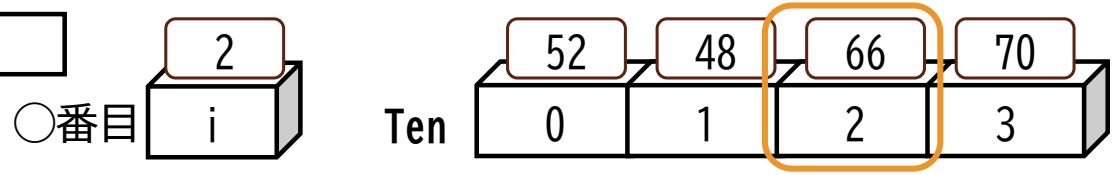


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48

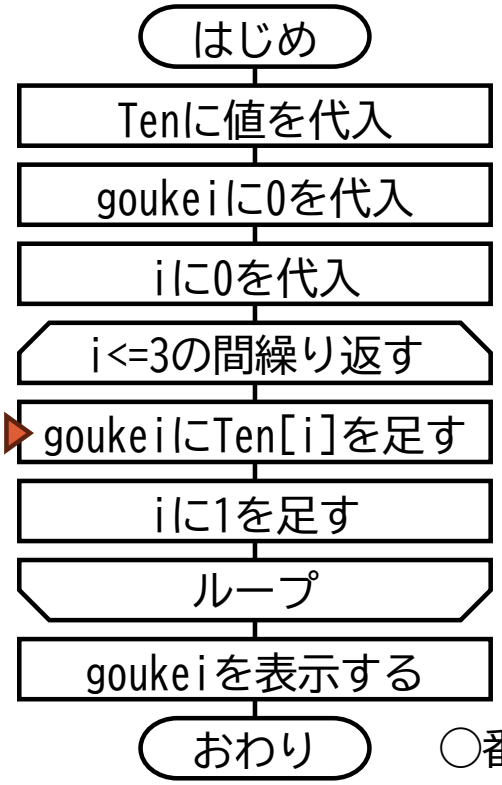
いまのTen[2]は66



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

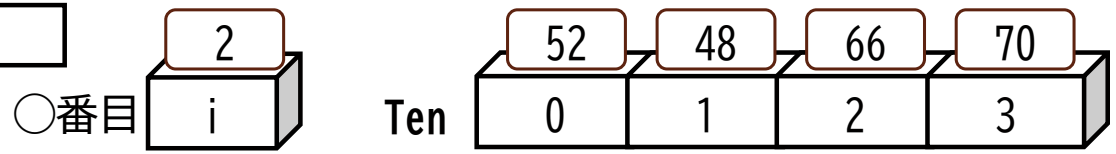


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48

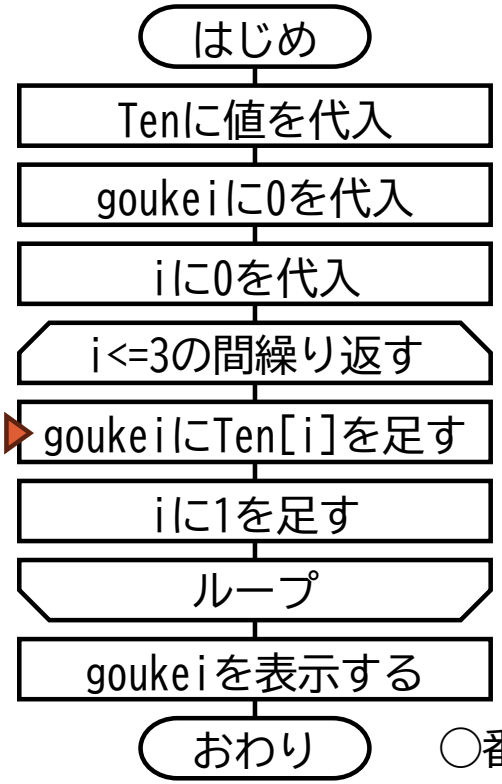
いまのTen[2]は66



配列は反復処理と組み合わせる

- 「＝」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

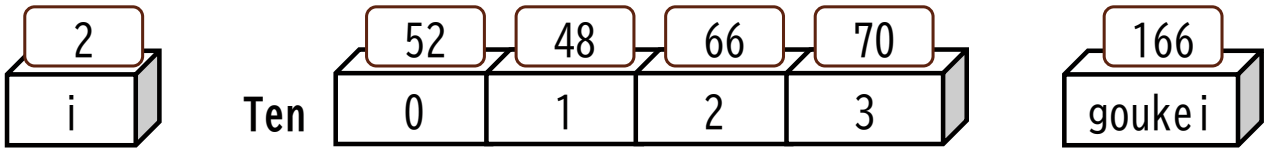
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

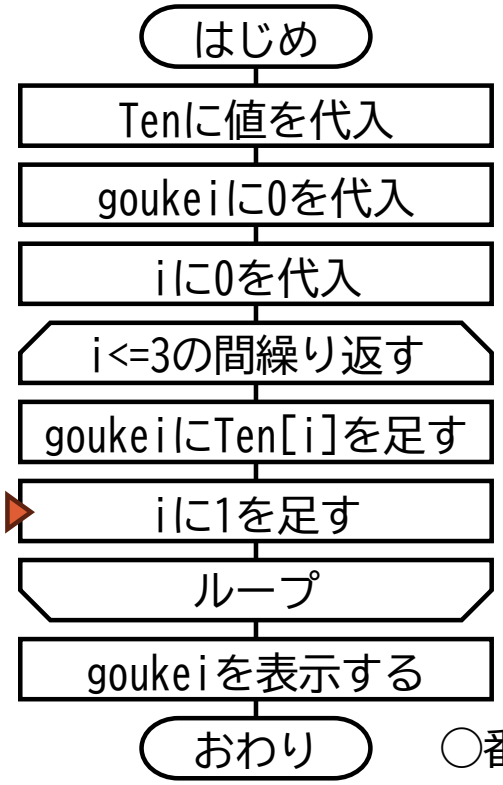
数学での計算
合計＝52＋48＋66



配列は反復処理と組み合わせる

- 「＝」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

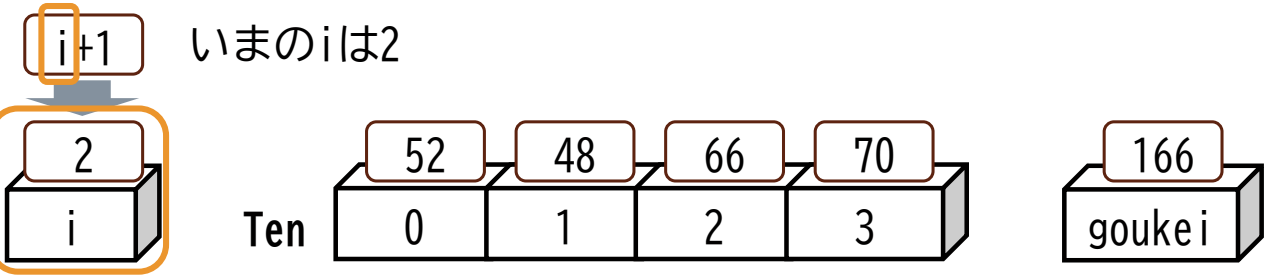
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

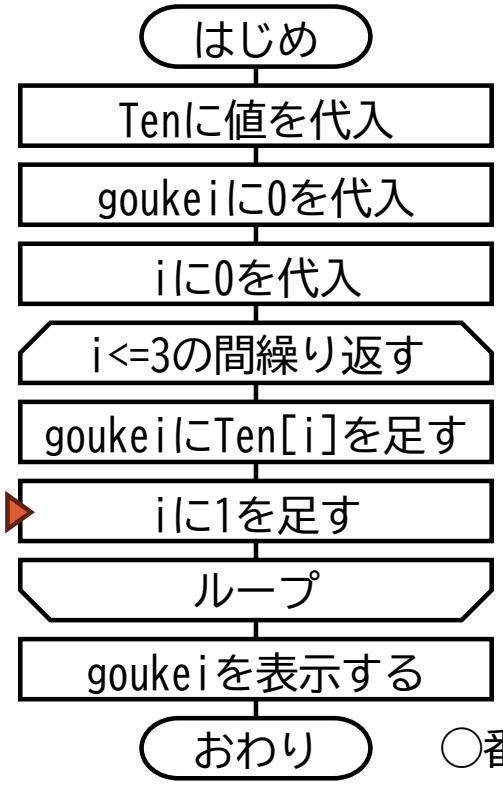
数学での計算
合計＝52＋48＋66



配列は反復処理と組み合わせる

- 「＝」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

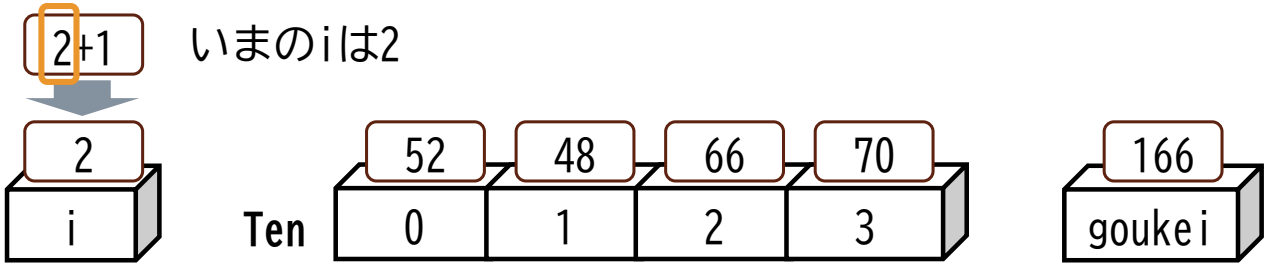
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

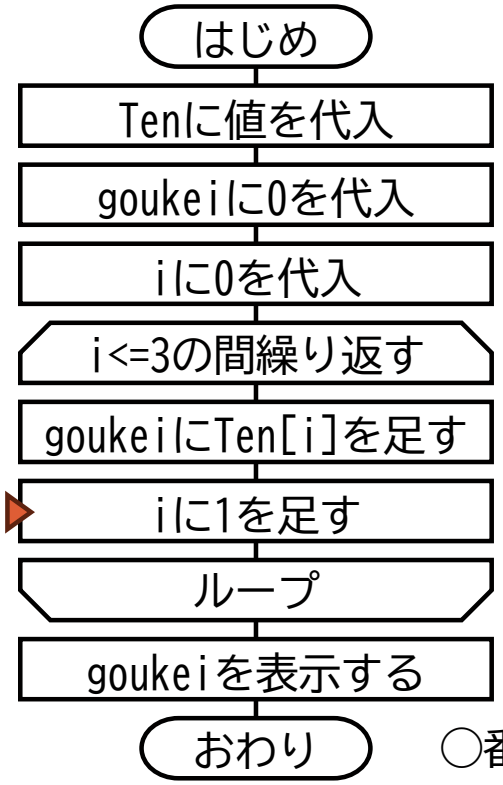
数学での計算
合計＝52＋48＋66



配列は反復処理と組み合わせる

- 「＝」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

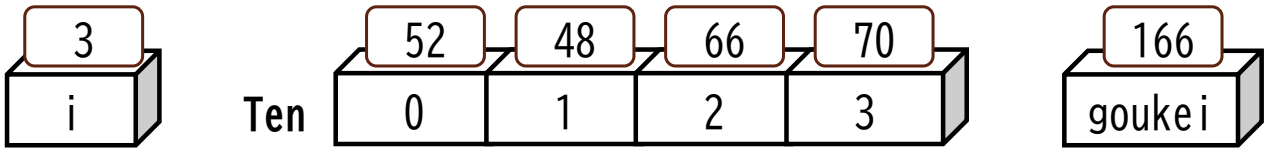
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

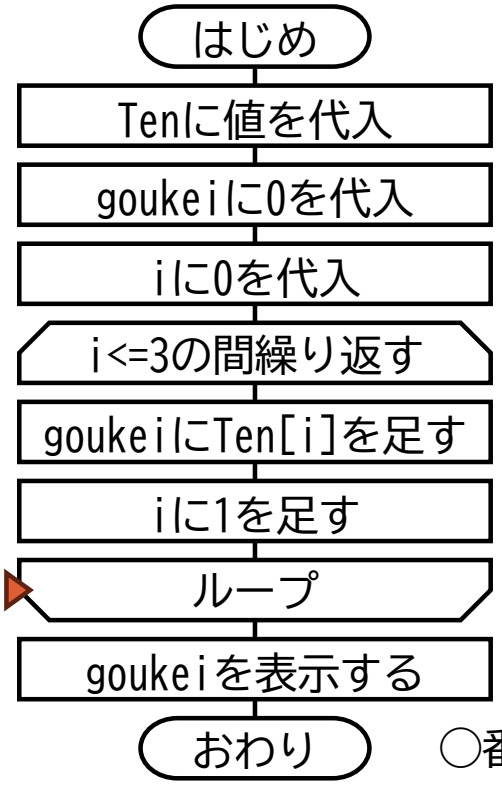
数学での計算
合計＝52＋48＋66



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

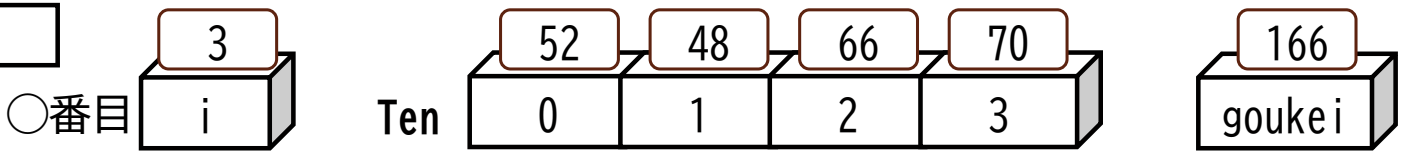


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

反復処理の終了部分を意味する

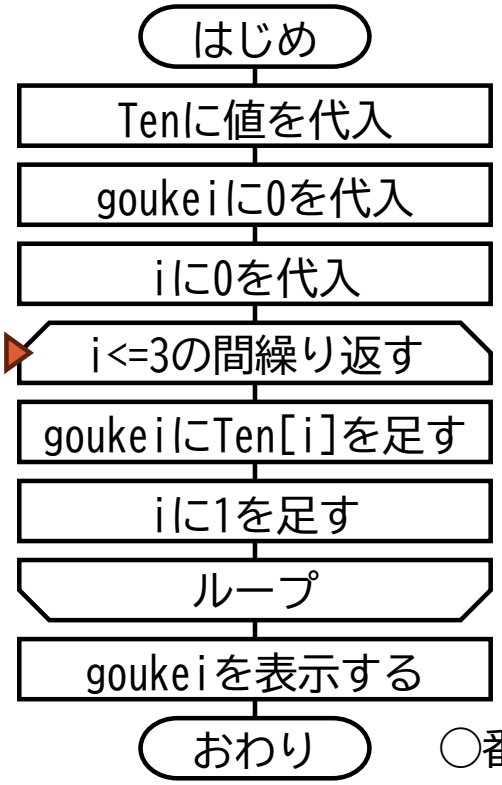
数学での計算
合計＝52＋48＋66



配列は反復処理と組み合わせる

- 「＝」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

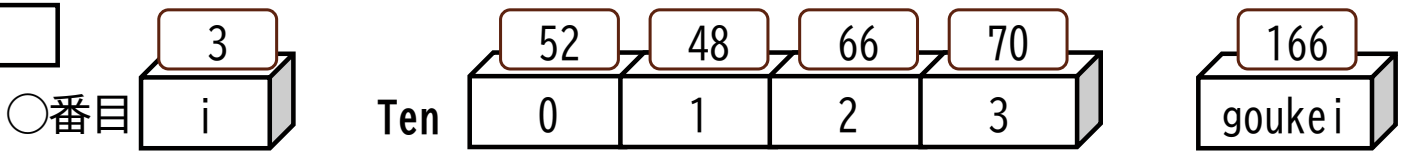
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

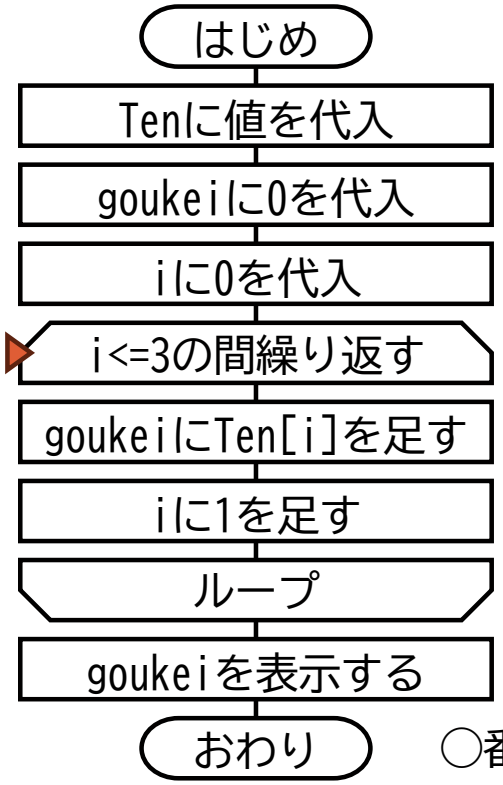
数学での計算
合計＝52＋48＋66



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

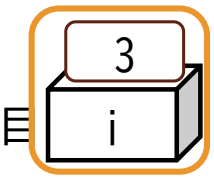


共通テストで使用されるDNCL

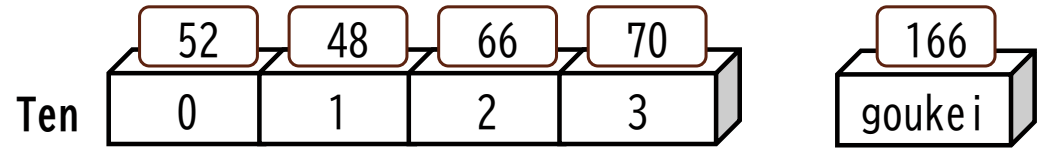
```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48＋66

いまのiは3で、条件を満たす



○番目

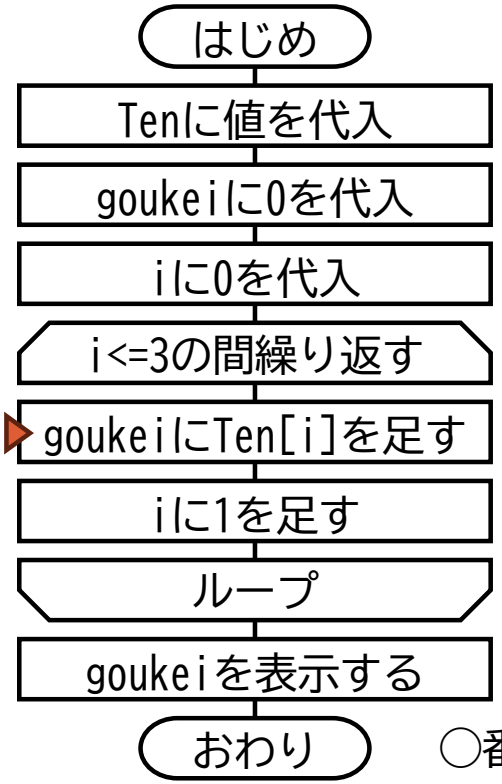


赤字：用語も意味も理解が必要、緑字：意味や概念の理解が必要

配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

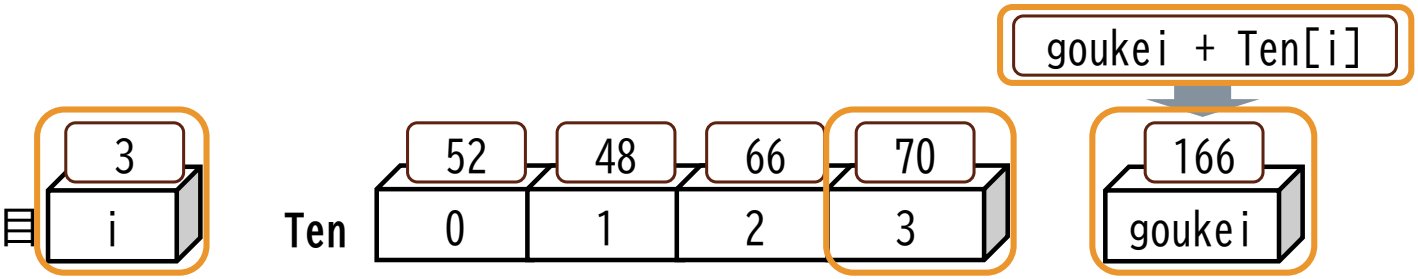


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48＋66

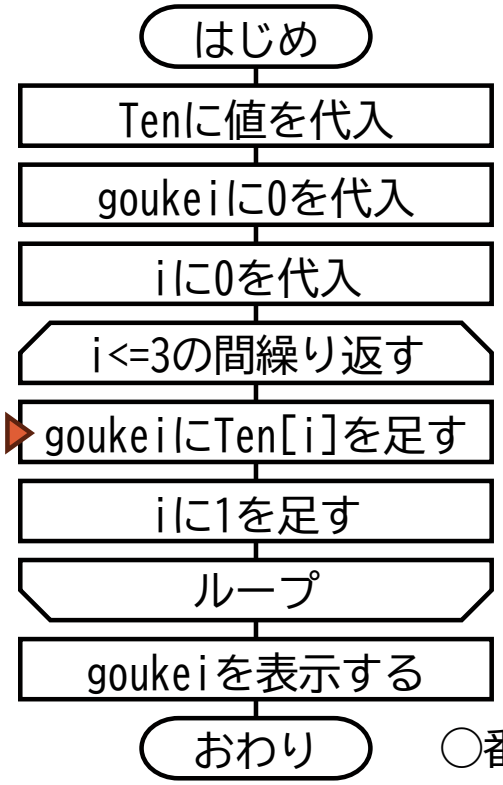
いまのgoukeiは166
いまのiは3
いまのTen[3]は70



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

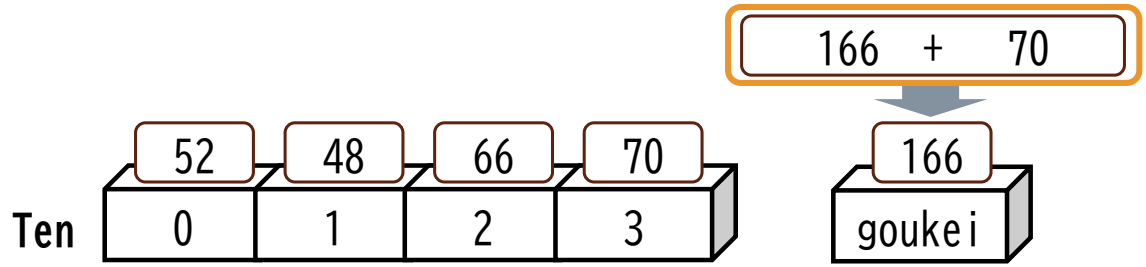


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

数学での計算
合計＝52＋48＋66

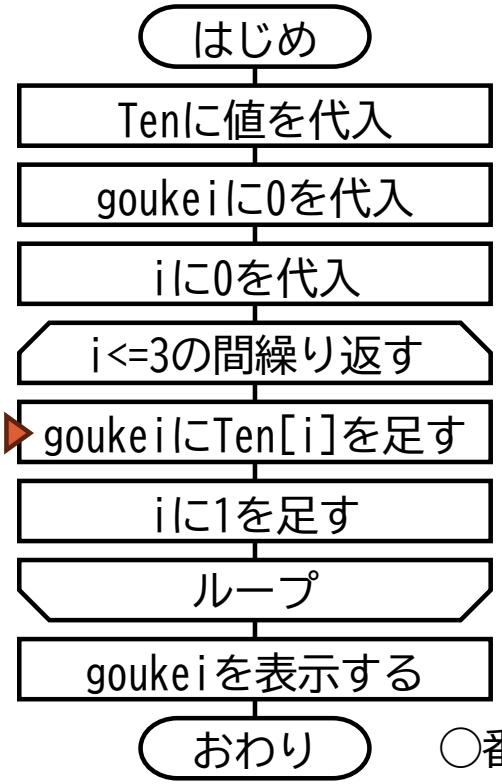
いまのgoukeiは166
いまのiは3
いまのTen[3]は70



配列は反復処理と組み合わせる

- 「＝」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

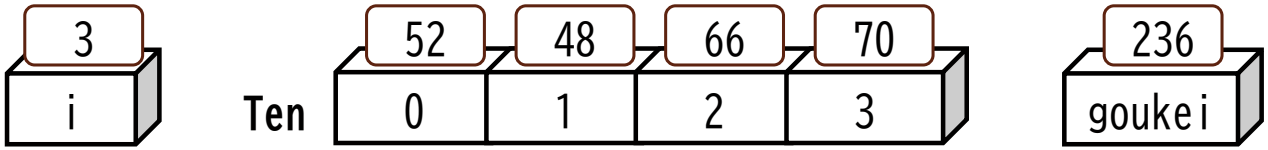
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

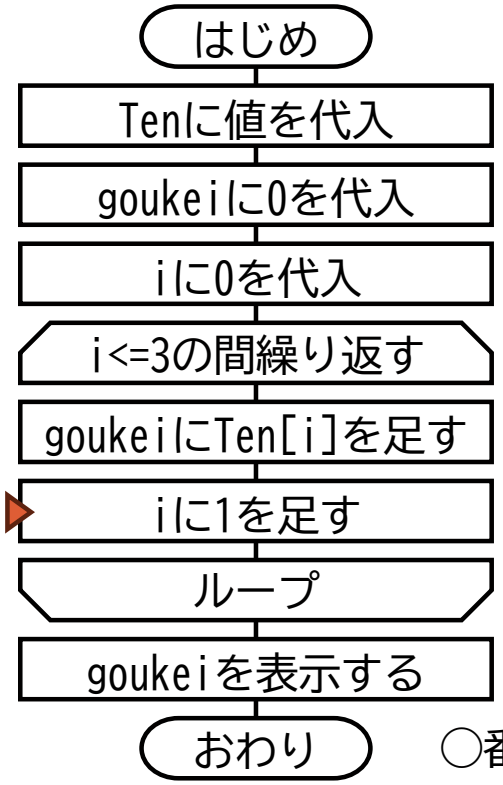
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

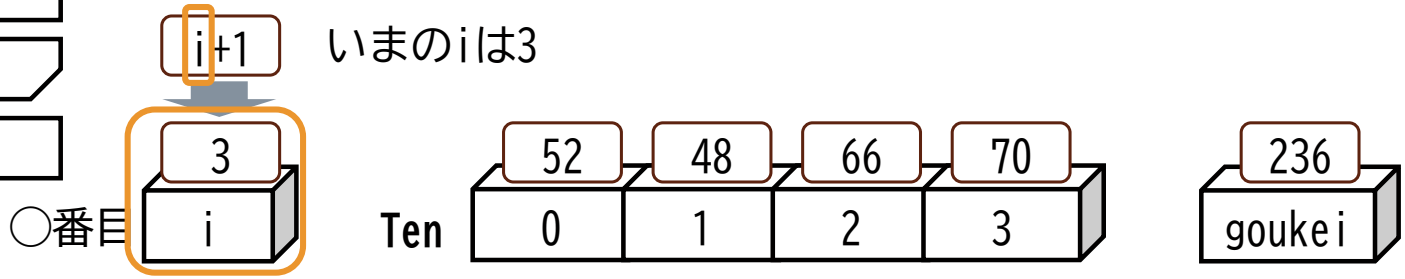
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

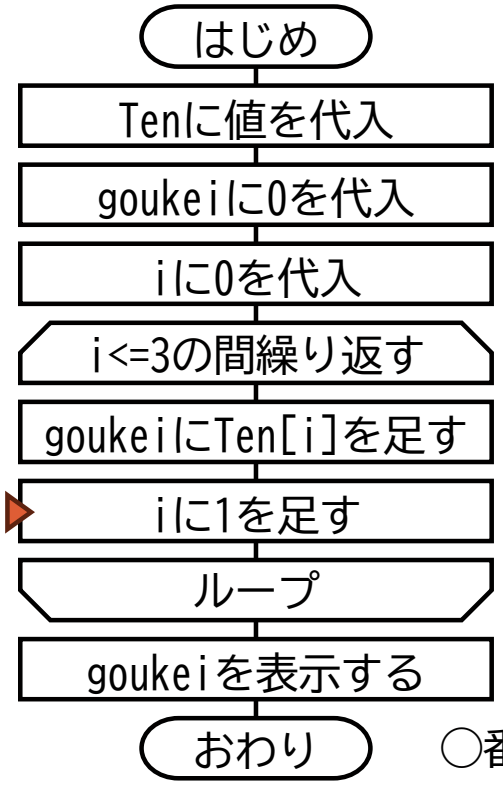
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

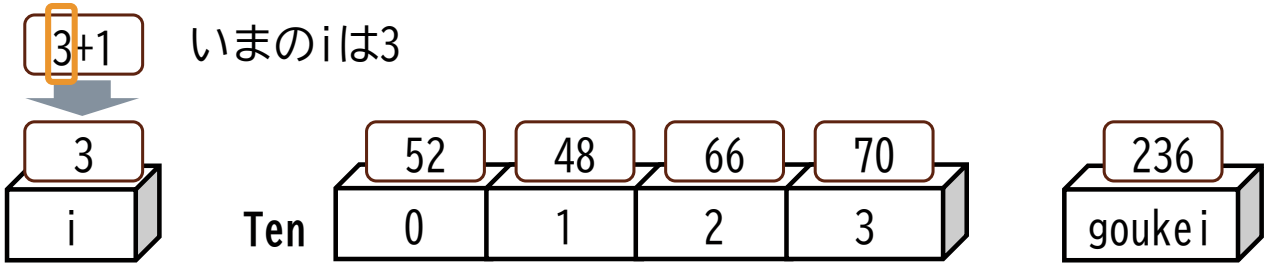
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

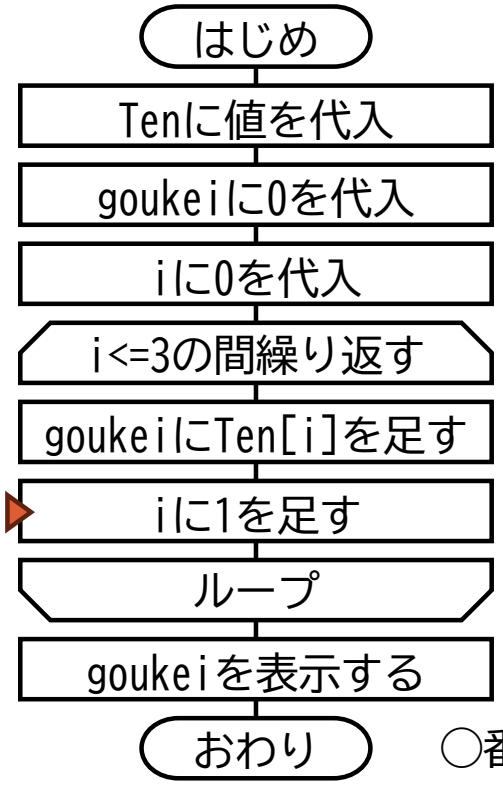
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

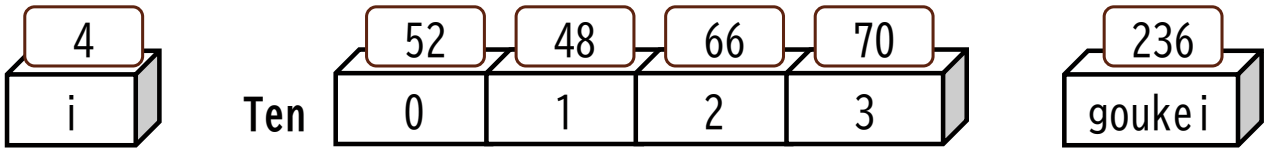
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

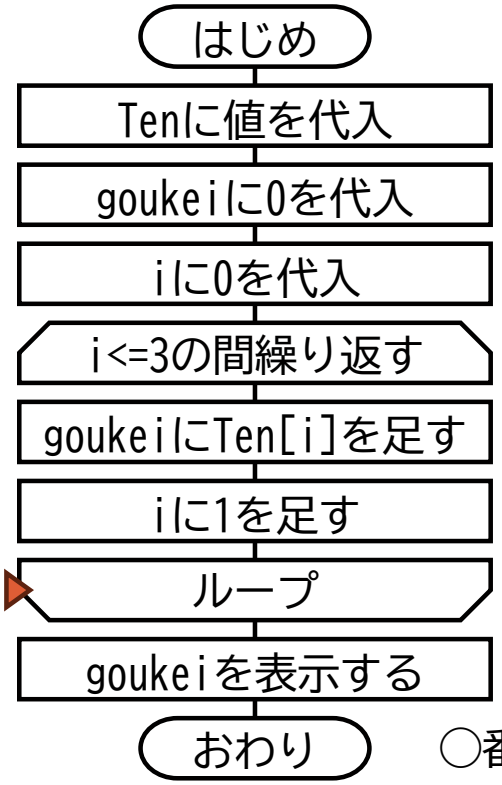
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

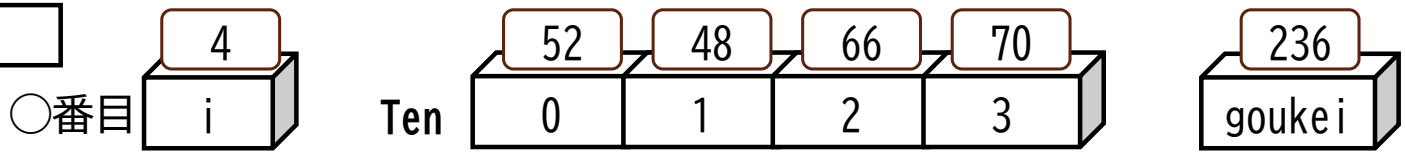


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

反復処理の終了部分を意味する

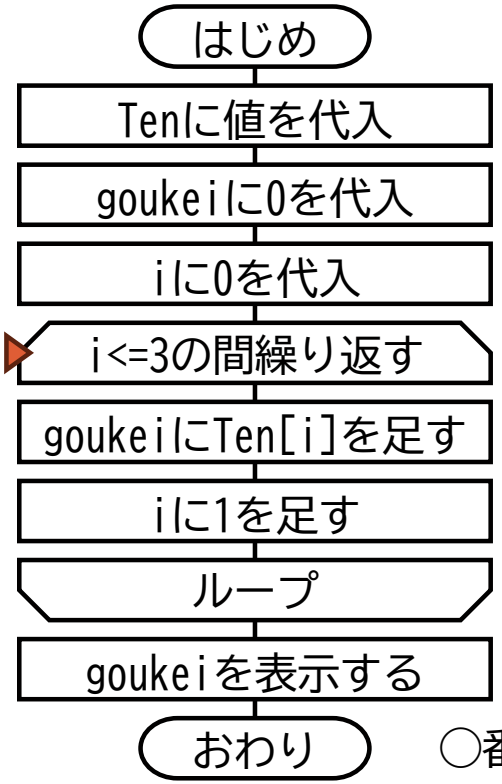
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

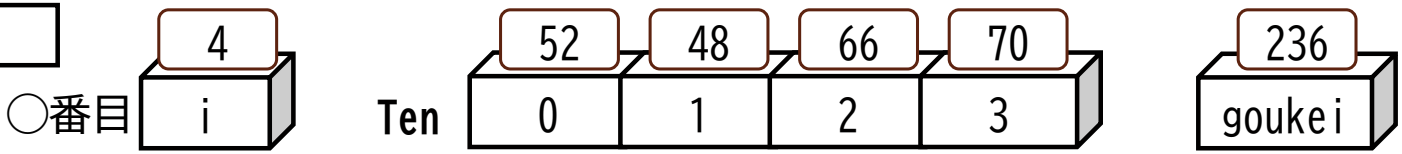
4科目の点数(52,48,66,70)の合計を求める



共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

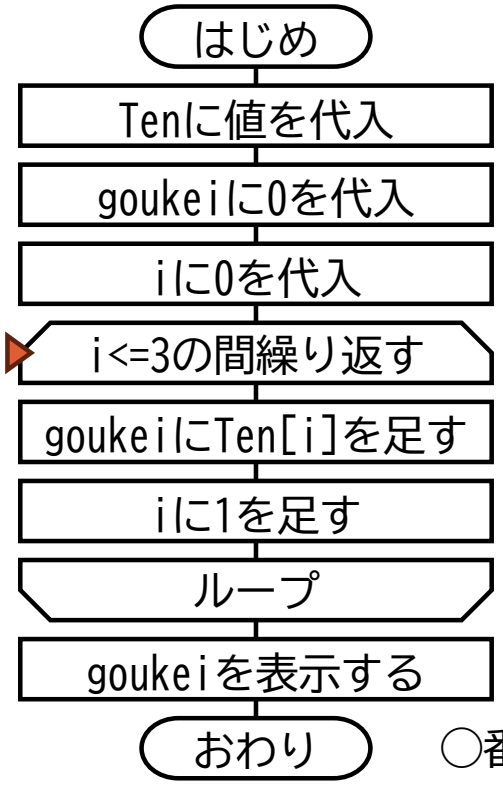
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

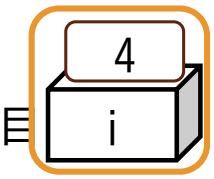


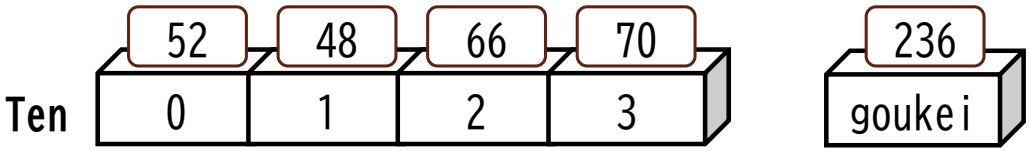
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

いまのiは4で、条件を満たさない

数学での計算
合計＝52＋48＋66＋70

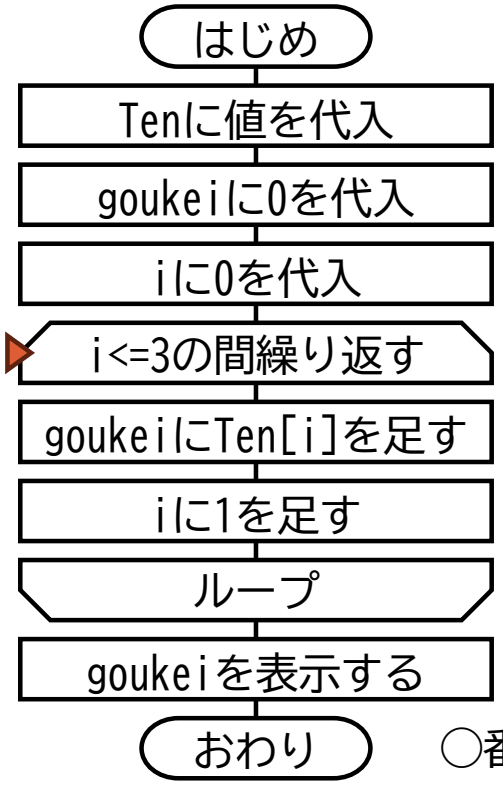




配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める



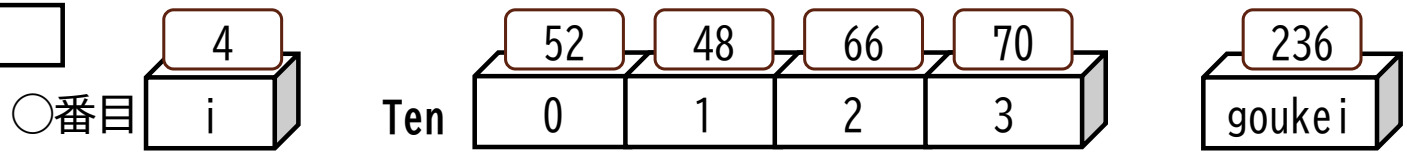
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

いまのiは4で、条件を満たさない

反復処理の終了部分の次に進む

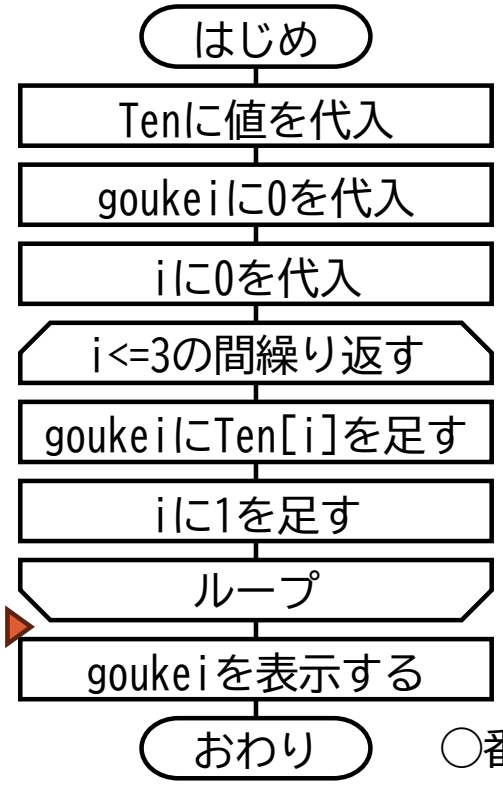
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

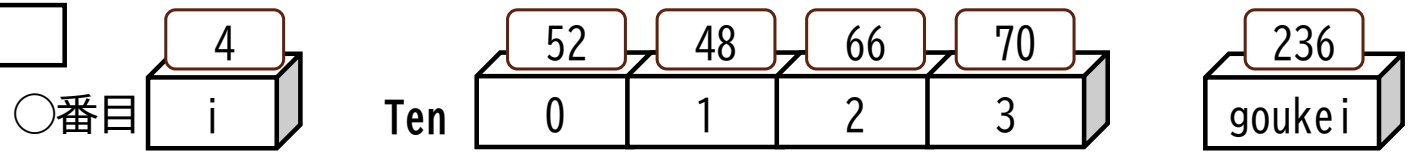


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

いまのiは4で、条件を満たさない

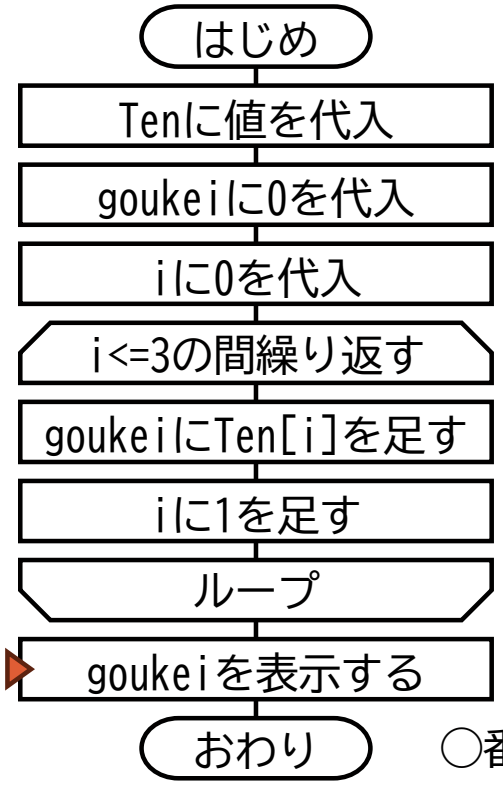
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める

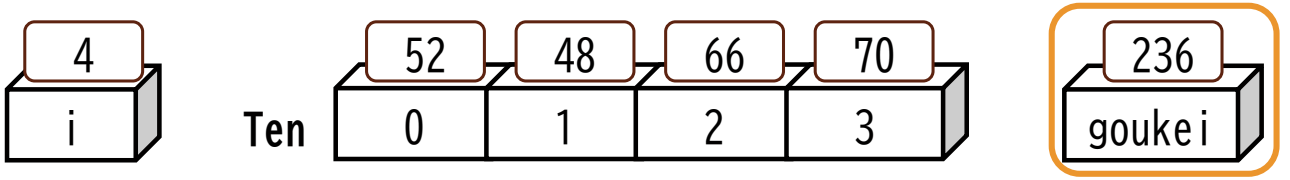


共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

いまのgoukeiは236

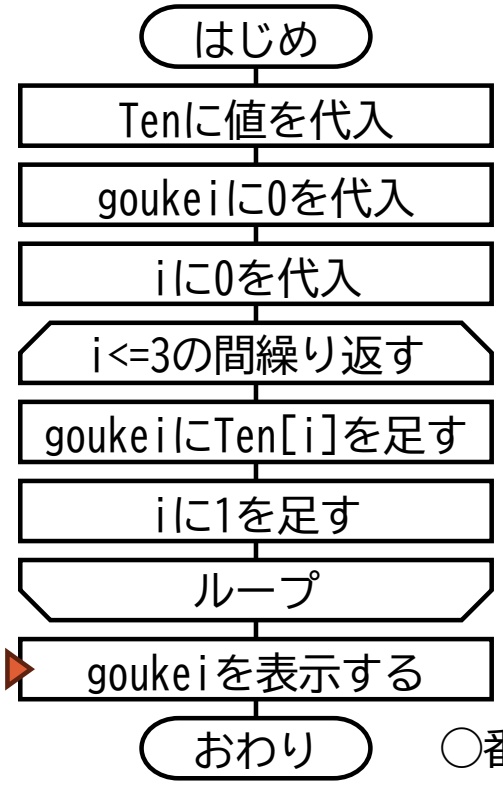
数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

- 「＝」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

4科目の点数(52,48,66,70)の合計を求める



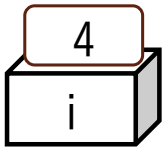
共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

「236」と表示される

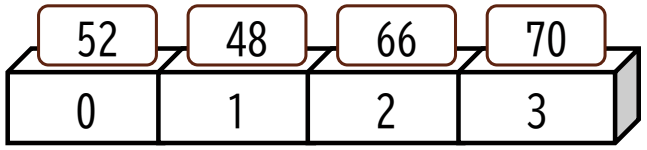
いまのgoukeiは236

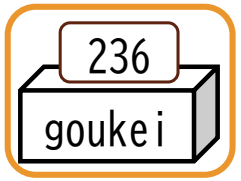
数学での計算
合計＝52＋48＋66＋70



○番目

Ten





配列は反復処理と組み合わせる

- 「＝」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

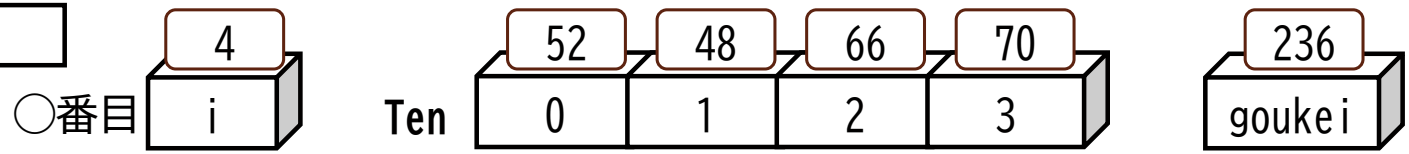
4科目の点数(52,48,66,70)の合計を求める

共通テストで使用されるDNCL

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
  goukei = goukei + Ten[i]
  i = i + 1
表示する(goukei)
```

「236」と表示される

数学での計算
合計＝52＋48＋66＋70



配列は反復処理と組み合わせる

4科目の点数(52,48,66,70)の合計を求める

配列を使用したプログラム

```
Ten = [52, 48, 66, 70]
goukei = 0
i = 0
i <= 3 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

配列を使用しないプログラム

```
Eigo = 52
Sugaku = 48
Kokugo = 66
Joho = 70
goukei = Eigo + Sugaku + Kokugo + Joho
表示する(goukei)
```

配列は反復処理と組み合わせることでプログラムをシンプルにできる

6科目の点数(52, 48, 66, 70, 58, 82)の合計を求めるプログラムに変更する

配列を使用したプログラム

```
Ten = [52, 48, 66, 70, 58, 82]
goukei = 0
i = 0
i <= 5 の間繰り返す：
    goukei = goukei + Ten[i]
    i = i + 1
表示する(goukei)
```

配列を使用しないプログラム

```
Eigo = 52
Sugaku = 48
Kokugo = 66
Joho = 70
XXXX = 58
YYYY = 82
goukei = Eigo + Sugaku + Kokugo + Joho + XXXX + YYYY
表示する(goukei)
```

配列を使用しない場合は、100個など数を増やすと対応しづらい
配列を使用していると、数が増えてもプログラムがシンプルになる

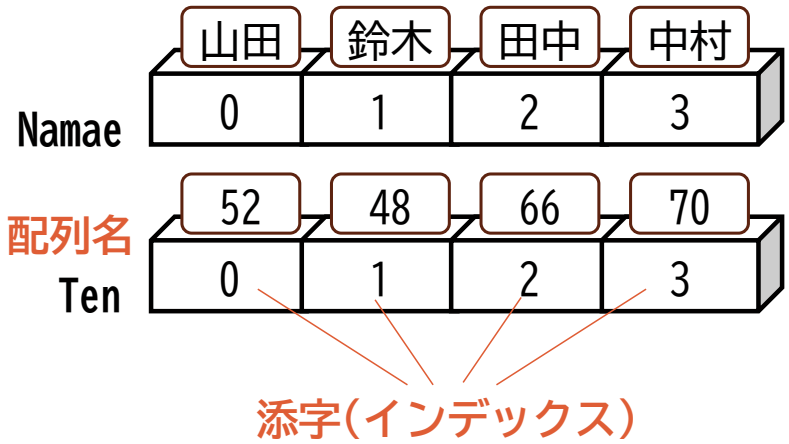
「プログラミング(配列)」の要点

「プログラミング(配列)」の要点

配列：複数の変数をまとめたもの

生徒	山田	鈴木	田中	中村
点数	52	48	66	70

イメージ



配列　Namae=[”山田”,”鈴木”,”田中”,”中村”]

Namae[0]=”山田”
Namae[1]=”鈴木”
Namae[2]=”田中”
Namae[3]=”中村”

配列名[添字]で
個々の値に
アクセスする

配列　Ten=[52, 48, 66, 70]

Ten[0]=52
Ten[1]=48
Ten[2]=66
Ten[3]=70

配列名[添字]で
個々の値に
アクセスする