



☆交換

右の 3 行のプログラムは、2 つの変数 a, b を ^{こうかん}交換するプログラムである。
このプログラム内で使われる変数 temp を ^{いちじへんすう}一時変数 (テンポラリ変数) という。

```
temp = a
a = b
b = temp
```

例題 1 出力結果を答えよ。

- (1) Tensu = [5, 3]
- (2) temp = Tensu[0]
- (3) Tensu[0] = Tensu[1]
- (4) Tensu[1] = temp
- (5) 表示する (Tensu)

T ₀	T ₁	temp
5	3	5
3	5	

出力結果: 3, 5

例題 2 出力結果を答えよ。

- (1) Tensu = [5, 3, 7, 1]
- (2) n = 要素数(Tensu)
- (3) j を 1 から n - 1 まで 1 ずつ増やしながら繰り返す:
- (4) | もし Data[j] < Data[0] ならば:
- (5) | | temp = Data[0]
- (6) | | Data[0] = Data[j]
- (7) | | Data[j] = temp
- (8) 表示する (Data)

j	T ₀	T ₁	T ₂	T ₃	Data[j] < Data[0]
	5	3	7	1	
1	3	5	7	1	3 < 5 O
2	3	5	7	1	7 < 3 x
3	1	5	7	3	1 < 3 O

出力結果: 1, 5, 7, 3

1 出力結果を答えよ。

- (1) Tensu = [6, 8]
- (2) temp = Tensu[0]
- (3) Tensu[0] = Tensu[1]
- (4) Tensu[1] = temp
- (5) 表示する (Tensu)

T ₀	T ₁	temp
6	8	6
8	6	

出力結果: 8, 6

2 出力結果を答えよ。

- (1) Tensu = [6, 8, 2, 4]
- (2) n = 要素数(Tensu)
- (3) j を 1 から n - 1 まで 1 ずつ増やしながら繰り返す:
- (4) | もし Data[j] < Data[0] ならば:
- (5) | | temp = Data[0]
- (6) | | Data[0] = Data[j]
- (7) | | Data[j] = temp
- (8) 表示する (Data)

j	T ₀	T ₁	T ₂	T ₃	Data[j] < Data[0]
	6	8	2	4	
1	6	8	2	4	8 < 6 x
2	2	8	6	4	2 < 6 O
3	2	8	6	4	4 < 2 x

出力結果: 2, 8, 6, 4

☆ソート

データをある規則にしたがって並べ替えることをソートという。

小さい値から大きな値へと並べ替える昇順^{しょうじゅん}と、大きな値から小さな値へと並べ替える降順^{こうじゅん}がある。

3 出力結果を答えよ。

例題 3 出力結果を答えよ。

- (1) Tensu = [5, 3, 7, 1]
- (2) n = 要素数(Tensu) 2 4
- (3) i を 0 から n 2 まで 1 ずつ増やしながら繰り返す：
- (4) | j を i + 1 から n 3 まで 1 ずつ増やしながら繰り返す：
- (5) | | もし Data[j] < Data[i] ならば：
- (6) | | | temp = Data[i]
- (7) | | | Data[i] = Data[j]
- (8) | | | Data[j] = temp
- (9) 表示する (Data)

- (1) Tensu = [6, 8, 2, 4]
- (2) n = 要素数(Tensu)
- (3) i を 0 から n - 2 まで 1 ずつ増やしながら繰り返す：
- (4) | j を i + 1 から n - 1 まで 1 ずつ増やしながら繰り返す：
- (5) | | もし Data[j] < Data[i] ならば：
- (6) | | | temp = Data[i]
- (7) | | | Data[i] = Data[j]
- (8) | | | Data[j] = temp
- (9) 表示する (Data)

i	j	T ₀	T ₁	T ₂	T ₃	Data[j] < Data[i]
0	1	5	3	7	1	3 < 5 0
	2	3	5	7	1	7 < 3 x
	3	1	5	7	3	1 < 3 0
1	2	7	5	7	3	7 < 5 x
	3	1	3	7	5	3 < 5 0
2	3	1	3	5	7	5 < 7 0

出力結果： 1, 3, 5, 7

i	j	T ₀	T ₁	T ₂	T ₃	Data[j] < Data[i]
0	1	6	8	2	4	8 < 6 x
	2	2	8	6	4	2 < 6 0
	3	2	8	6	4	4 < 2 x
1	2	2	6	8	4	6 < 8 0
	3	2	4	8	6	4 < 6 0
2	3	2	4	6	8	6 < 8 0

出力結果： 2, 4, 6, 8