



## ☆交換

右の3行のプログラムは、2つの変数 a,b を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 を一時変数 (テンポラリ変数) という。

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

※配列の添字は0から始まるものとする。要素数(Data)は、配列の要素数を返す関数である。

例題1 出力結果を答えよ。

- (1) a = 5, b = 1
- (2) temp = a
- (3) a = b
- (4) b = temp
- (5) 表示する ("a の値は", a, "b の値は", b)

出力結果: a の値は    b の値は   

a	b	temp
5	1	5
1	5	

1 出力結果を答えよ。

- (1) a = 6, b = 8
- (2) temp = a
- (3) a = b
- (4) b = temp
- (5) 表示する ("a の値は", a, "b の値は", b)

出力結果: a の値は    b の値は   

a	b	temp
6	8	6
8	6	

例題2 出力結果を答えよ。

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

出力結果:   

j	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	5	1	7	3
1	1	5	7	3
2	1	5	3	7

2 出力結果を答えよ。

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

出力結果:   

j	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	6	8	2	4
1	6	8	2	4
2	6	2	8	4
3	6	2	4	8

3 出力結果を答えよ。

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

出力結果: 最小値は   

j	m	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	0	6	8	2	4
1	1				
2	2				
3	3				

## ☆最小値

(3) ~ (6) は、  
0番目からn-1番目の  
最小値を求めるプログラム。

☆バブルソート

配列の隣り合うデータの大小を比較して交換することを繰り返す手法を、バブルソートという。  
繰り返し文の中にさらなる繰り返し文があるプログラムを、二重ループという。

例題4 出力結果を答えよ。

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

出力結果: 1, 3, 5, 7

i	j	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
1	0	5	1	7	3
	1	1	5	7	3
	2	1	5	3	7
	3	1	3	5	7

☆選択ソート

要素の中から最小（最大）のものを見つけ出し、先頭にもっていくことを繰り返す手法を、選択ソートという。

例題5 出力結果を答えよ。

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

出力結果: 1, 3, 5, 7

i	min	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	1	1	5	9	3
	3	1	3	7	5
	2	3	1	3	5
	3	1	3	5	7

4 出力結果を答えよ。

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

出力結果: 2, 4, 6, 8

i	j	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
1	0	6	8	2	4
	1	6	8	2	4
	2	6	2	8	4
	3	0	2	6	8

5 出力結果を答えよ。

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

出力結果: 2, 4, 6, 8

i	min	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	2	2	8	6	4
	1	3	2	4	8
	2	2	2	4	6
	3	0	2	4	8