

☆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
```

※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 1 bの値は 5

a	b	temp
5	1	5
1	5	

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： 1, 5, 3, 7

j	D ₀	D ₁	D ₂	D ₃
	5	1	7	3
0	1	5	7	3
1	1	5	7	3
2	1	5	3	7



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 8 bの値は 6

a	b	temp
6	8	6
8	6	

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： 6, 2, 4, 8

j	D ₀	D ₁	D ₂	D ₃
	6	8	2	4
0	6	8	2	4
1	6	2	8	4
2	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 4

j	m	D ₀	D ₁	D ₂	D ₃
		6	8	2	4
1					
2	2				
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^{にじゅう}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:
- (5) | | もし Data[j] > Data[j+1] ならば:
- (6) | | | temp = Data[j]
- (7) | | | Data[j] = Data[j+1]
- (8) | | | Data[j+1] = temp
- (9) 表示する (Data)

出力結果:

1, 3, 5, 7

i	j	D ₀	D ₁	D ₂	D ₃
		5	1	7	3
1	0	1	5	7	3
	1	1	5	7	3
	2	1	5	3	7
2	0	1	5	3	7
	1	1	3	5	7
3	0	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^{せんたく}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:
- (6) | | もし Data[min] > Data[j] ならば:
- (7) | | | min = j
- (8) | temp = Data[i]
- (9) | Data[i] = Data[min]
- (10) | Data[min] = temp
- (11) 表示する (Data)

出力結果:

1, 3, 5, 7

i	min	D ₀	D ₁	D ₂	D ₃
		5	1	7	3
0	1	1	5	7	3
1	3	1	3	7	5
2	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:
- (5) | | もし Data[j] > Data[j+1] ならば:
- (6) | | | temp = Data[j]
- (7) | | | Data[j] = Data[j+1]
- (8) | | | Data[j+1] = temp
- (9) 表示する (Data)

出力結果:

2, 4, 6, 8

i	j	D ₀	D ₁	D ₂	D ₃
		6	8	2	4
1	0	6	8	2	4
	1	6	2	8	4
	2	6	2	4	8
2	0	2	6	4	8
	1	2	4	6	8
3	0	2	4	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:
- (6) | | もし Data[min] > Data[j] ならば:
- (7) | | | min = j
- (8) | temp = Data[i]
- (9) | Data[i] = Data[min]
- (10) | Data[min] = temp
- (11) 表示する (Data)

出力結果:

2, 4, 6, 8

i	min	D ₀	D ₁	D ₂	D ₃
		6	8	2	4
0	2	2	8	6	4
1	3	2	4	6	8
2	2	2	4	6	8