

以下の疑似言語で書かれたソートアルゴリズムを C++ で実装しなさい。
また、それぞれのソートの種類を判別しなさい。

○関数 OUT(配列名、配列数)
は、配列の要素を 1 行に列挙する関数です。

(表示例)

7 6 2 8 4 5 3 1 9

○Sort①(A[], N)
○int N: 配列の要素数
○int A: 0～(N-1) の配列

```
Sort①(A, N)
■ for i ← 1 to N-1
| • v ← A[i]
| • j ← i - 1
| ■ j >= 0かつA[j] > v
| | • A[j+1] ← A[j]
| | • j--
| ■
| • A[j+1] = v
| • OUT (A, N)
■
```

○Sort②(A[], N)
○void swap(X, Y): 配列中の X と Y の要素を交換する操作
○int N: 配列の要素数
○int A: 0～(N-1) の配列

```
Sort②(A[] N)
■ for i ← 0 to N-1
| • minj ← i
| ■ for j ← i to N-1
| | ▲ if A[j] < A[minj]
| | | • minj ← j
| | ▼
| ■
| • swap(A[i], A[minj])
| • OUT (A, N)
■
```

○Sort③(A[], N)
○void swap(X, Y): 配列中の X と Y の要素を交換する操作
○int N: 配列の要素数
○int A:

```
0～(N-1) の要素を持つ配列
○bool flag:
繰り返し制御のためのフラグ

Sort③(A[], N)
flag ← true
■ flag = false
| • flag ← false
| ■ for j ← N-1 to 1
| | ▲ A[j] < A[j-1]
| | | • swap(A[j], A[j-1])
| | | • flag ← true
| | ▼
| ■
| • OUT (A, N)
■
```

実行結果

結果①

7 6 2 8 4 5 3 1 9

結果②

7 6 2 8 4 5 3 1 9

結果③

7 6 2 8 4 5 3 1 9