応用練習問題

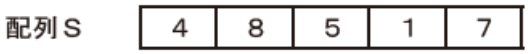
▼Practice01

ソートアルゴリズムの一つである挿入法についての説明を読み、これに基づいたプログラムを作成してください。

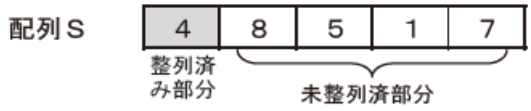
［流れ図の説明］

以下に示すような手順で一次元配列Ｓを昇順に並べ替えるものである。

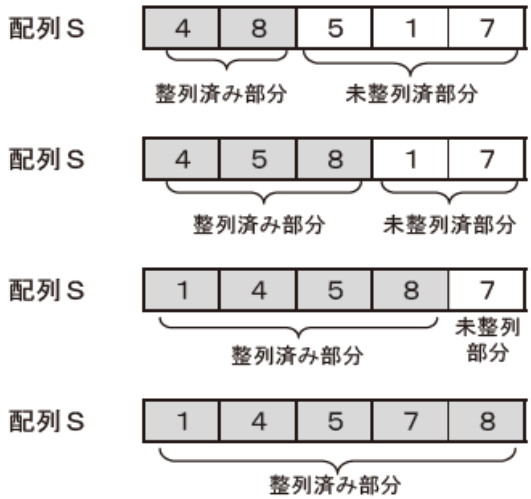
［手順］



① 最初の段階では，整列されていないため，最初の要素だけを整列済みの要素と考える。



② 未整列部分の要素を，先頭から順番に 整列済みの部分の適切な位置に挿入する。



③ 未整列部分の要素がなくなったら，整列処理を終了する。

実行結果

```
ソート前>　５２７１４
ソート後>　１２４５７
```

▼Practice02

以下の説明を読み、これに基づいたプログラムを作成してください。

※クラス名は「Practice02」とし、配列 A,B はそれぞれ図の通り要素数 10 の配列としてコードの中で定義すること。

［プログラムの説明］

配列A（A[0], A[1], ･･･, A[n－1]）に格納された数値の降順に付けた順位を，配列Bに求めるプログラム ranking1 である。

① プログラム ranking1 は，データが格納された整数型の１次元配列A，配列Aの要素数n，順位を格納する整数型の１次元配列Bを引数として受け取る。

② プログラム ranking1 では，数値が同じ場合は要素番号（添字の値）の小さい方を高い（小さい）順位とする。

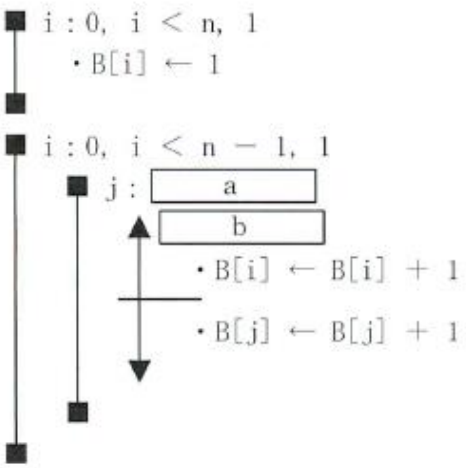
③ プログラム ranking1 の順位付けの例を，以下に示す。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
配列A	65	37	48	92	37	54	20	65	37	81
配列B	3	7	6	1	8	5	10	4	9	2

［プログラム］

○ranking1(整数型: A[], 整数型: n, 整数型: B[])

○整数型: i, j



aに関する解答群

ア 0, j ＜ i, 1　　イ 0, j ＜ n, 1

ウ i ＋ 1, j ＞ 0, －1　　エ i ＋ 1, j ＜ n, 1

bに関する解答群

ア A[i] ＞ A[j]　　イ A[i] ≧ A[j]

ウ A[i] ≦ A[j]　　エ A[i] ＜ A[j]

実行結果

```
[0]: 65 点　3 位
[1]: 37 点　7 位
…（略）
[9]: 81 点　2 位
```

↑実際の実行結果では[0]～[9]すべて表示させてください

▼Practice03

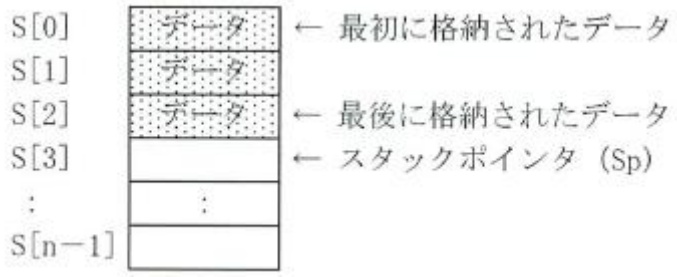
以下の説明と実行例を参考に、スタックを操作するプログラムを作成してください。

尚、スタックのサイズは 10 個とし、メニュー画面で 0 が入力されるまでループし続けます。

push →スタックにデータを積む（サイズを超える場合は「スタックが満杯です」と表示する）

pop　→スタックからデータを取り出して表示する（データが何もない場合は「スタックが空です」と表示する）

スタックの出力　→現在のスタックを画面に出力する（　　〃　　）



実行例 1 （いきなり終了）

```
0: 終了
1: push
2: pop
3: スタックの出力
0

プログラムを終了します
```

実行例 2 （スタックが空のパターン）

```
0: 終了
1: push
2: pop
3: スタックの出力
3
スタックは空です

0: 終了
1: push
2: pop
3: スタックの出力
2
スタックは空です
※実際はこの後も 0 が入力されるまでループし続けます
```

実行例 3（スタック→表示→ポップ→表示）

```
0: 終了
1: push
2: pop
3: スタックの出力
1
push するデータを入力してください
10

0: 終了
1: push
2: pop
3: スタックの出力
1
push するデータを入力してください
50

0: 終了
1: push
2: pop
3: スタックの出力
3
[0] 10
[1] 50

0: 終了
1: push
2: pop
3: スタックの出力
2
50 を取り出しました

0: 終了
1: push
2: pop
3: スタックの出力
3
[0] 10

※実際はこの後も 0 が入力されるまでループし続けます
```

他に、スタックを 10 個以上積もうとしたときに「スタックが満杯です」と表示されるか確認してください。