

- ファイル名のXXXXXXXXXのところは学籍番号に変更してください
- プログラム中にstudent_numberに自分の学籍番号を入れてください

```
student_number = XXXXXXXXX
```

課題2: バブルソート

バブルソートはソートアルゴリズムの1種である。隣り合う2つの要素を比較し、必要であれば交換する操作を繰り返し行い、交換の必要がなくなるまで行うことによって昇順あるいは降順に要素をソートする。

例として、以下のようなリストをバブルソートで昇順にソートする場合を考えてみよう

[10, 5, 3]

1. 先頭の10と5を比較。昇順になっていないので交換する -> [5, 10, 3]
2. 右にずれて、今度は10と3を比較。やはり昇順になっていないので交換 -> [5, 3, 10]
3. ここでリストの最後まで来たので、もう1度先頭に戻る -> [5, 3, 10]
4. 先頭の5と3を比較。昇順になっていないので交換 -> [3, 5, 10]
5. 右にずれて、5と10を比較。昇順になっているのでそのまま -> [3, 5, 10]
6. リストの最後まで来たので先頭に戻る -> [3, 5, 10]
7. 先頭の3と5を比較。昇順なのでそのまま -> [3, 5, 10]
8. 右にずれて5と10を比較。昇順なのでそのまま -> [3, 5, 10]
9. リストの最後まで来たが、今回はリストの先頭から最後まで処理する間に1回も交換が発生しなかったなのでここでソートは終了とする -> [3, 5, 10]

従って今回のバブルソートの結果は [3, 5, 10] となる

では「引数に整数のリストを与えると、そのリストをバブルソートで昇順にソートして返り値に返す関数」を実装しよう。ただし今回は採点の都合上、「要素の交換が起きるたびにリストをprintする」仕様とする

実行例:

```
> bubble_sort([10, 5, 3])
[5, 10, 3]
[5, 3, 10]
[3, 5, 10]
# 教員コメント: bubble_sort()の返り値は[3, 5, 10]となっていないといけない

> bubble_sort([2, 1])
[1, 2]
# 教員コメント: bubble_sort()の返り値は[1, 2]となっていないといけない

> bubble_sort([3])
```

```
[3]
# 教員コメント: bubble_sort()の返り値は[3]と (以下略)

> bubble_sort([])
[]
# 教員コメント: bubble_sort()の返り値は[]と (以下略)
```

ちなみに

- 引数は整数のリスト以外の場合は考えなくて良いものとする
- 引数のリストの長さが0の場合はありうるものとする
- 引数で与えたリストをそのまま変更しても、コピーして使っても構わないものとする。ただしどちらにしても返り値でソートしたリストを返すこと
- リストのprint方法だが、例えばリストの変数を `foo_list` とすると、単純に `print(foo_list)` とすれば良い (その他の方法でも構わないが、余計な空白やタブなどが入ってしまうのは避けること)

また、使うかもしれないヒントとして「リストを添字付きでループする」方法は例えば以下の2通りが存在する

```
num_array = [1, 2, 3]

# パターン1
for index in range(len(num_array)):
    print(index, num_array[index])

# パターン2
for index, x in enumerate(num_array):
    print(index, num_array[index], x) # ここでnum_array[index]とxは同じものとなる
```

```
def bubble_sort(num_array):

    """整数のリストをバブルソートする関数
    Args:
        num_array (list[int]): ソートしたいリスト

    Returns:
        list[int]: ソートした結果のリスト
    """
```

