

a) passed by value이기 때문에 값을 넘기긴 하지만 함수 진행 후에 변수의 값이 변하지 않는다.

swap(value, list[0]) : value = 2 / list[5] = {1, 3, 5, 7, 9}

swap(list[0], list[1]) : value = 2 / list[5] = {1, 3, 5, 7, 9}

swap(value, list[value]) : value = 2 / list[5] = {1, 3, 5, 7, 9}

b) passed by reference이기 때문에 주소값이 넘어가므로 함수 진행 후 변수의 값이 변화한다.

swap(value, list[0]) : value = 1 / list[5] = {2, 3, 5, 7, 9}

swap(list[0], list[1]) : value = 1 / list[5] = {3, 2, 5, 7, 9}

swap(value, list[value]) : value = 2 / list[5] = {3, 1, 5, 7, 9}

c) passed by value-result이기 때문에 값을 넘긴 후에 함수가 진행됨에 따라 값이 변화한 후 반환된다.

swap(value, list[0]) : value = 1 / list[5] = {2, 3, 5, 7, 9}

swap(list[0], list[1]) : value = 1 / list[5] = {3, 2, 5, 7, 9}

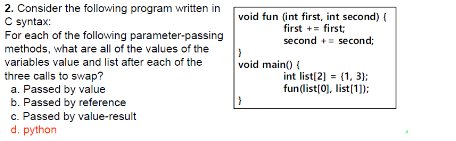
swap(value, list[value]) : value = 2 / list[5] = {3, 1, 5, 7, 9}

d) python은 immutable한 변수에 대하여 passed by value, mutable한 변수에 대하여 passed by reference의 성질을 띈다.

swap(value, list[0]) : value = 2 / list[5] = {1, 3, 5, 7, 9}

swap(list[0], list[1]) : value = 2 / list[5] = {1, 3, 5, 7, 9}

swap(value, list[value]) : value = 2 / list[5] = {1, 3, 5, 7, 9}



a) fun(list[0], list[1]) : list[2] = {1, 3}

b) fun(list[0], list[1]) : list[2] = {2, 6}

c) fun(list[0], list[1]) : list[2] = {2, 6}

d) fun(list[0], list[1]) : list[2] = {1, 3}

1번과 풀이과정 동일