# PL_HW5_20191583_김태곤

1. a. passed by value

| | value | list |
|---|---|---|
| swap(value, list[0]) | 2 | {1,3,5,7,9} |
| swap(list[0], list[1]) | 2 | {1,3,5,7,9} |
| swap(value, list[value]) | 2 | {1,3,5,7,9} |

swap이 실행될 때 actual parameter 값이 formal parameter 값으로 복사된다. swap이 종료된 후, (subprogram 마무리된 후) 아무것도 변하지 않는다.

b. passed by reference

| | value | list |
|---|---|---|
| swap(value, list[0]) | 1 | {2,3,5,7,9} |
| swap(list[0], list[1]) | 1 | {3,2,5,7,9} |
| swap(value, list[value]) | 2 | {3,1,5,7,9} |

actual parameter의 주소값을 넘겨준다. 변경 사항이 유지된다.

c. passed by value-result

| | value | list |
|---|---|---|
| swap(value, list[0]) | 1 | {2,3,5,7,9} |
| swap(list[0], list[1]) | 1 | {3,2,5,7,9} |
| swap(value, list[value]) | 2 | {2,1,5,7,9} |

시작할 때 initialize 하고 끝나면 copy back 한다. 즉 시작할 때 pass by value로 값을 넘겨고, 종료되면 callee의 값이 caller로 전달된다.

d. python (pass-by-assignment)

| | value | list |
|---|---|---|
| swap(value, list[0]) | 2 | {1,3,5,7,9} |
| swap(list[0], list[1]) | 2 | {1,3,5,7,9} |
| swap(value, list[value]) | 2 | {1,3,5,7,9} |

원래의 값은 안바뀐다. 객체 내부는 바꿀 수 있지만 객체 자체는 못바꾼다. 새로운 오브젝트가 형성된다.

2. a. passed by value

초기 list[2] = {1,3}

calling fun(list[0], list[1]) → list[2] = {1,3}

1번 설명처럼 sub program이 종료된 후 아무것도 안변한다.

b. passed by reference

초기 list[2] = {1,3}

calling fun(list[0], list[1]) → list[2] = {2,6}

→ list[0] = list[0] + list[0] = 2

list[1] = list[1] + list[1] = 6

주소값을 넘겨서 변경사항이 유지된다.

c. passed by value-result

calling fun(list[0], list[1]) → list[2] = {2,6}

위 설명과 같다. (1번)

d. python

calling fun(list[0], list[1]) → list[2] = {2,6}