

프로그래밍 언어론

과제 #03

1. Value model of variables를 따르는 C언어로 아래 코드를 작성했다. (1)은 deep 또는 shallow assignment인지 답하고, 아래 코드에서 나타날 수 있는 문제와 해결 방법이 무엇인지 간략하게 설명하라

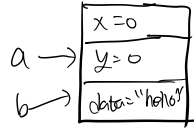
```
typedef struct {
    int x, y;
    char* data
} A;
A a = { 0, 0, NULL };
a.data = (char*) malloc(100);
strcpy(a.data, "hello"); /* 문자열 복사 */
A b = a;                  /* (1) */
free(a.data);
printf("x = %d, y = %d, data = %s\n", b.x, b.y, b.data);
```

1.

1) 위의 예는 shallow object 같아 보일다.

왜냐하면 A a = {0, 0, Null} 에서 data에 해당하는 "hello"를 표시한 구조체 a를
A 구조체의 b 변수에 assign을 했으므로 같은 데이터를 가리킨다.

즉



2) 위에서 생성할 수 없는 문제는 a.data가 free 되었으므로, b의 data를 참조할 수 없으므로
printf에서 출력되지 않는다.

해결방법은 A b

```
A a = {0, 0, Null};
```

```
a.data = (char*) malloc(100);
```

```
strcpy(a.data, "hello");
```

```
A b;
```

```
b.x = a.x;
```

```
b.y = a.y;
```

```
b.data = (char*) malloc(100);
```

```
strcpy(a.data, b.data);
```

2.

```
1. type cell (* a forward declaration *) (* 타입 선언 *)
2. type cell_ptr = pointer to cell (* 타입 선언 *)
3. x : cell (* 변수 선언 *)
4. type cell = record (* 타입 선언 *)
5.     val : integer (* 타입 선언 *)
6.     next : cell_ptr (* 타입 선언 *)
7. y : cell (* 변수 선언 *)
```

앞에서 보인 코드에서 4 번째 줄에 있는 내용은 앨리어스(alias)를 만드는 것인가? strict name equivalence를 사용하는 경우에, x와 y는 같은 자료형이라고 볼 수 있는가?

- (* *)는 주석 표시, 빨간색 글씨는 줄 번호

- record는 구조체를 선언하는 것으로 보면 됨. 4 번째 줄에 있는 record는 val과 next를 필드(field)로 포함하는 구조체임

2

1. 4번째 줄에 있는 내용이 alias 인가?

필요불필요를 모르겠다 예과 header가 같은 공간을 가리키는 alias이다.

2. x와 y는 다른 변수이다.

alias를 만든다고 해도 이는 strict name equivalence 예외이다.