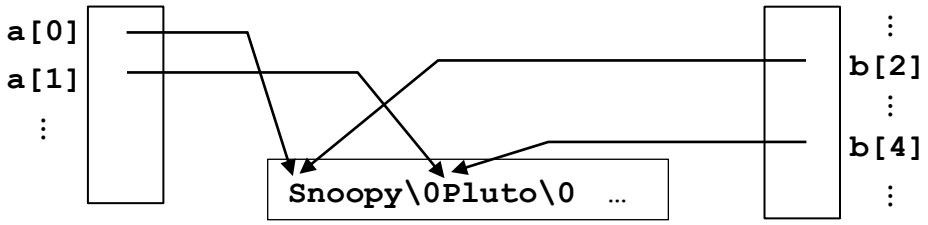
Homework #4 notification

The sample test contains a test case on C-style strings, as replicated below.

```
const char* a[6]={"Snoopy","Pluto","Garfield","Shrek",
                                              "Micky","Lorax"};
const char* b[7]={"Garfield","Doraamon","Snoopy","Micky",
                                    "Pluto","Shrek","Lorax"};
test_lcs(a,6,b,7);
```

To find an lcs of two sequences of C-style strings, we ought to use **strcmp** to compare two C-style strings for equality. However, for this particular test case, we may use == instead, because the strings here are usually shared in a string pool, as mentioned in the last semester.
For example, **a[0]** and **b[2]** share the string **"Snoopy"**, **a[1]** and **b[4]** share the string **"Pluto"**, and so on. Thus, the address equality **a[0]==b[2]** implies the string equality **strcmp(a[0],b[2])==0**.



string pool

Of course, we should use **strcmp** in general case. For example, with the declarations
const char a[6][9]={*same as above***};**
const char b[7][9]={*same as above***};**
the strings aren't shared and we can't use == for string equality.

However, to focus on dynamic programming, we shall not involve in handling the general case in this homework.