APS106: FUNDAMENTALS OF COMPUTER PROGRAMMING
LAB 9 - MONDAY, MARCH 31, 2:00 - 4:00

OBJECTIVE:

At this stage in the course, you've had lot's of experience with looping through arrays using array notation ([ ]). This lab takes this a step further, and asks that you do much the same thing with pointer notation, and pointer arithmetic.

The only array notation that should appear in your code this week is in the initial declaration of arrays and strings.

The only libraries you may include are stdio.h and stdlib.h.

PROBLEM:

Write a program that asks a user for two keyboard inputs: (i) a string A of less than 80 characters, and (ii) a string B of less than 80 characters. Print out the two strings. Then pass the two strings to a separate function, called first_instance(), that searches string A for the first occurrence of string B. If B is not found within A, first_instance() returns -1. If B is found within A, then the function returns the offset from the beginning of A where B begins. Finally, main() interprets the return code, and prints an appropriate message.

For example, input might look like this:

```
enter a string A (< 80 chars):  a ab abc abcd abcde
enter a string B (< 80 chars):  abcd
```

and output would look like this:

```
string A:  a ab abc abcd abcde
string B:  abcd
string B offset 9 chars from beginning of A
```

NOTES:

Within the function first_instance, it may be helpful to know the lengths of A and B. Ordinarily, you'd use the strlen() function, but that's only available through string.h. Recognize that it may be a good idea to immediately calculate the lengths yourself, via a simple for or while loop.

The rest of the function is more challenging to write. There are likely many ways to do this. But consider writing two loops: one that moves through A one character at a time, and a second loop inside the first that compares characters in A from that point on to characters from the start of B.