# Computer Science I Sample Program: Substring Search

## Please check Webcourses for the Due Date

## Read all the pages before starting to write your code

A substring is a contiguous subsection of a string. For example, "put" is a substring of "computer", since the characters starting at index 3 and ending at index 5 in "computer" are exactly the string "put".

**The Problem**

Given a string and a substring with in to search for, you must determine, if it exists, the starting index of **the first occurrence** of the substring in the given string. You must also determine if no such index exists.

**The Input (to be read from standard input)**

The first line will contain a single positive integer, $c$ ($c \leq 25$), representing the number of test cases to process. The test cases follow.

Each test case will be two lines long. The first line of each test case will contain a single string, ***s***, of lower case letter. The second line of each test case will contain a single string, ***t***, of lower case letters. Both ***s*** and ***t*** will be in between 1 and 99 characters, inclusive. ($1 \leq |\boldsymbol{s}|, |\boldsymbol{t}| \leq 99$)

**The Output (to be printed to standard out)**

For each input case, output a single integer on a line by itself: the starting index (0 based) of the first occurrence of the substring ***t*** inside the string ***s***, or -1, if ***t*** is not a substring of ***s***.

**Sample Input**

```
3
computer put
abaaabaaabaaa aa
coffee tea
```

**Sample Output**

```
3
2
-1
```

**Implementation Restrictions/ Run-Time/Memory Restrictions**

1. You must read the deck for each case into statically allocated string(s).

2. For full credit, your algorithm must run in O(|***s***|*|***t***|) time. This means that you can only do a set of nested loops: an outer loop through the characters in the first string and an inner loop through the characters of the second string.

3. For full credit, you must have appropriately designed functions. **In particular, any correct solution which is fully coded in main will NOT receive full credit.**

3. You must only declare your string variables **INSIDE** your case loop.

**Deliverables**

You must submit one file over WebCourses:

1) A source file, *substring.c*. Please use stdin, stdout. There will be an automatic 10% deduction if you read input from a file or write output to a file.

2) A file describing your testing strategy, *lastname_Testing.doc(x) or lastname_Testing.pdf*. This document discusses your strategy to create test cases to ensure that your program is working correctly. If you used code to create your test cases, just describe at a high level, what your code does, no need to include it.

3) Files *substring.in* and *substring.out*, storing both the test cases you created AND the corresponding answers, respectively.