

Testing Document for Substring Program (by Arup Guha)

After the sample, I chose a string (baseball) and then chose every suffix of the string as the substring for which to search for to make sure that different length substrings at the last spot worked. Then I added each prefix of the same string to see if the first possible match always works. Next, to check different positions, I took the same word and chose every 3 letter substring of that word. Since I had already tried some of these cases, I just put in the new substrings (the middle 4) that I hadn't tried. In each of these cases, I could eyeball the correct answer. The only tricky case was "baseball" and "l" where the answer is 6 and not 7 since the first "l" appears in index 6. (I made it a case because that letter appears in index 7.)

Since the first substring is supposed to be reported and so far there was only one test case where a string contained more than one copy of a substring (baseball with l), I wanted to test this more thoroughly. I created a test case with all a's so that the shorter string with all a's appears in many places. Then I created another test case with rotating letters a, b and c and checked for the substring "bcabc" which occurs many times in the larger string of rotating letters.

Since the input specification didn't prevent the substring to be longer than the search string (so this case should have -1 as an answer), I tried a simple case like this where the substring was "aa" and the search string was only "a".

To test the one letter case at the end, I created a case where the last letter is unique, so "computer" and "r".

Finally I put a random case using letters a, b and c where my substring matches the first few letters in many cases but messes up most of the time. Here is that case:

abcabc**abcabca**abacaba abcabcaa

I added a couple cases where the answer is -1 since I had so few. In these cases I came within 1 letter of a valid match.

Last but not least I wanted to add both a minimum and maximum test case. Two min test cases are where both strings are length 1, one with an answer of 0, the other with an answer of -1. A max case has a string of size 99 for the search string and both a string of size 99 or smaller for the substring.

While there are other cases I could think about, this seems to round out lots of standard cases where the substring is various lengths, and at various locations in the target string, including the beginning and the end, and also where the substring appears multiple times in the main string.