CS13002 Programming and Data Structures, Spring 2005, Section 4/D

Laboratory Test II **Total points: 25** **March 22, 2005**

For students with odd PC numbers

This exercise deals with pattern matching in strings. Let A, B, C be given strings. We plan to find the pattern $B * C$ in A. Here $*$ stands for any substring. So the pattern $B * C$ means the occurrence of the string B followed by any string (possibly empty) followed in turn by the string C. As an example, consider the following strings:

```
A = "Dashing through the snow on a one-horse open sleigh"
B = "now on a one"
C = "pen"
```

The pattern $B * C$ exists in A:

```
Dashing through the snow on a one-horse open sleigh
                     <---------><------><->
                           B        *     C
```

On the other hand, search fails with A and B as above but for the following values of C:

```
C = "Jingle"      (C is not at all a substring of A)
C = "rough"       (C comes earlier than B in A)
C = "on"          (No occurrence of C strictly after the only occurrence of B in A)
```

Write a program that does the following:

- Read three strings A, B and C.
- Report if the pattern $B * C$ is present in A.
- If the search is successful, also report the start index of a match.

Use static character arrays to store the strings A, B, C. A function that returns the index of the leftmost match of a string T in a string S may be helpful for your program. Note that the word *substring* precludes the possibility of gaps in the matching. For example, `horses` and `tough` are not substrings of A in the above example.

Report the output of your program for the following test cases:

```
   A = "What fun it is to ride and sing a sleighing song tonight"

a) B = "fun and sing"         C = "song"
b) B = "it is to rid"         C = "e and s"
c) B = "night"                C = ""
d) B = ""                     C = "tonight"
e) B = "to ride and sing"     C = "it is"
f) B = "sleighing"            C = "hi"
g) B = "g"                    C = "g"
```

CS13002 Programming and Data Structures, Spring 2005, Section 4/D

Laboratory Test II **Total points: 25** **March 22, 2005**

For students with <u>even</u> PC numbers

This exercise deals with pattern matching in strings. Let A, B, C be given strings and n a non-negative integer. We plan to find the pattern $B.\{n\}C$ in A. Here . stands for a <u>single</u> character and $\{n\}$ implies exactly n occurrences. So the pattern $B.\{n\}C$ means the occurrence of the string B followed by any sequence of exactly n characters followed in turn by the string C. As an example, consider the following:

```
A = "Dashing through the snow on a one-horse open sleigh"
B = "rough the"
C = "on"
n = 6
```

The pattern $B.\{n\}C$ exists in A:

```
Dashing through the snow on a one-horse open sleigh
          <-------><----><>
              B      n   C
```

For the above A, B, C match also occurs for $n = 11$. For no other values of n a match occurs. As another example, take A as above, but $B =$ `"on"`, $C =$ `"or"` and $n = 3$. Though the leftmost match of B in A does not correspond to the pattern $B.\{n\}C$, the second match of B in A does.

Write a program that does the following:

- Read three strings A, B and C and a non-negative integer n.
- Report if the pattern $B.\{n\}C$ is present in A.
- If the search is successful, also report the start index of a match.

Use static character arrays to store the strings A, B, C. A function that returns the index of the leftmost match of a string T in a string S may be helpful for your program. Note that the word *substring* precludes the possibility of gaps in the matching. For example, `horses` and `tough` are not substrings of A in the above example.

Report the output of your program for the following test cases:

```
   A = "What fun it is to ride and sing a sleighing song tonight"

a) B = "hat fun"          C = "is to"          n = 4
b) B = "ing"              C = "ong"            n = 15
c) B = "g"                C = "g"              n = 14
d) B = "unit"             C = "rid"            n = 7
e) B = "son"              C = "nights"         n = 4
f) B = "igh"              C = ""               n = 10
g) B = ""                 C = "hat"            n = 2
```