Question 3

Due on 2019-08-29, 23:59 IST
You are given a sequence of lower case alphabetical characters as input, say A[0], ..., A[N-1]. N will be at most 100.

A prefix of the sequence is some initial segment of the string - that is, A[0], ..., A[K-1], for some integer K, 1 <= K <= N.

You have to output the longest prefix which occurs multiple times in the string.

If no prefix occurs again, you must output "NO". (without quotes)

For example, if the input string is "ababaa", then the possible prefixes are {"a", "ab", "aba", "abab", "ababa", "ababaa"}. We can see that "a" occurs at indices 2, 4, and 5. The string "ab" occurs again in position 2. "aba" occurs in position 2. "abab" does not occur again in the string. So the longest prefix that occurs multiple times in the string is "aba".

**Sample Test Cases**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case 1</td>
<td>abanpa</td>
<td>a</td>
</tr>
<tr>
<td>Test Case 2</td>
<td>abcabcabc</td>
<td>abcabc</td>
</tr>
<tr>
<td>Test Case 3</td>
<td>abaa</td>
<td>a</td>
</tr>
<tr>
<td>Test Case 4</td>
<td>abz</td>
<td>NO</td>
</tr>
<tr>
<td>Test Case 5</td>
<td>ababaa</td>
<td>aba</td>
</tr>
<tr>
<td>Test Case 6</td>
<td>abr</td>
<td>NO</td>
</tr>
</tbody>
</table>

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.
Sample solutions (Provided by instructor)
```c
#include <stdio.h>

int compare ( char string [], int begin2, int length )
{
    int i1=0;
    int i2=begin2;
    for ( i1=0; i1<length; i1++, i2++ ) {
        if ( string [i1] != string [i2] ) {
            return 0;
        }
    }
    return 1;
}

int main()
{
    int i;
    char string[100];
    int string_length;
    int prefix_length;
    int max_prefix_length=0;
    gets(string);
    for ( i=0; string[i] != '\0'; i++ ) {
        ;
    }
    string_length=i;
    for ( prefix_length=1; prefix_length< string_length-1; prefix_length++ ) {
        for ( i=1; i < string_length-prefix_length+1; i++ ) {
            if ( compare ( string, i, prefix_length ) == 1 ) {
                max_prefix_length = prefix_length;
            }
        }
    }
    if ( max_prefix_length == 0 ) {
        printf ( "NO" );
    } else {
        for ( i=0; i<max_prefix_length; i++ ) {
            putchar ( string[i] );
        }
    }
    return 0;
}
```