2. Write a program to the Knuth-Morris-Pratt (KMP) string searching algorithm:

```
#include <stdio.h>
#include <string.h>
// Function to compute the Longest Prefix Suffix (LPS) array
void computeLPSArray(char* pattern, int m, int* lps) {
    int length = 0; // Length of the previous longest prefix suffix
    lps[0] = 0; // lps[0] is always 0
    int i = 1;

    while (i < m) {
        if (pattern[i] == pattern[length]) {
            length++;
            lps[i] = length;
            i++;
        } else {
            if (length != 0) {
                length = lps[length - 1];
            } else {
                lps[i] = 0;
                i++;
            }
        }
    }
}

// Function to implement the KMP string searching algorithm
void KMPSearch(char* pattern, char* text) {
    int m = strlen(pattern);
    int n = strlen(text);
    // Create the LPS array
    int lps[m];
    computeLPSArray(pattern, m, lps);

    int i = 0; // Index for text
    int j = 0; // Index for pattern

    while (i < n) {
        if (pattern[j] == text[i]) {
            i++;
            j++;
        }

        if (j == m) {
            printf("Pattern found at index %d\n", i - j);
            j = lps[j - 1];
        } else if (i < n && pattern[j] != text[i]) {
            if (j != 0) {
                j = lps[j - 1];
            } else {
                i++;
            }
        }
    }
}

// Main function
int main() {
    char text[] = "ABABDABACDABABCABAB";
    char pattern[] = "ABABCABAB";
    KMPSearch(pattern, text);
    return 0;
}
```