```
#include <stdio.h>

#define MAX_LENGTH 100

// Function to find the length of a string
int stringLength(const char str[]) {
    int length = 0;
    while (str[length] != '\0') {
        length++;
    }
    return length;
}

// Function to check if a string contains another string
int containsPattern(const char str[], const char pattern[]) {
    int i, j;

    for (i = 0; str[i] != '\0'; i++) {
        j = 0;
        while (pattern[j] != '\0' && str[i + j] == pattern[j])
{
            j++;
        }

        if (pattern[j] == '\0') {
            // Pattern found at index i in the main string
            return i;
        }
    }

    // Pattern not found
    return -1;
}

// Function to replace a pattern in a string
void replacePattern(char str[], const char pattern[], const
char replace[]) {
    int patternIndex = containsPattern(str, pattern);

    while (patternIndex != -1) {
        // Shift characters to the right to make space for
replacement
        int i;
        for (i = stringLength(str); i >= patternIndex +
stringLength(pattern); i--) {
```

```
            str[i + stringLength(replace) -
stringLength(pattern)] = str[i];
        }

        // Copy the replacement string into the main string
        for (i = 0; i < stringLength(replace); i++) {
            str[patternIndex + i] = replace[i];
        }

        // Check for the next occurrence of the pattern
        patternIndex = containsPattern(str, pattern);
    }
}

int main() {
    char mainString[MAX_LENGTH];
    char pattern[MAX_LENGTH];
    char replace[MAX_LENGTH];

    // Input main string, pattern, and replacement
    printf("Enter the main string: ");
    fgets(mainString, MAX_LENGTH, stdin);

    printf("Enter the pattern to replace: ");
    fgets(pattern, MAX_LENGTH, stdin);

    printf("Enter the replacement string: ");
    fgets(replace, MAX_LENGTH, stdin);

    // Remove newline characters from inputs
    mainString[stringLength(mainString) - 1] = '\0';
    pattern[stringLength(pattern) - 1] = '\0';
    replace[stringLength(replace) - 1] = '\0';

    // Perform pattern matching and replacement
    int patternIndex = containsPattern(mainString, pattern);
    if (patternIndex != -1) {
        replacePattern(mainString, pattern, replace);
        printf("Pattern found and replaced:\n%s\n",
mainString);
    } else {
        printf("Pattern not found in the main string.\n");
    }

    return 0;
```