**PROGRAM 2**

1. **Develop a Program in C for the following operations on Strings.**

**a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)**

**b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR .**

**Support the program with functions for each of the above operations. Don't use Built-in functions.**

Program:

#include <stdio.h>

#include <string.h>

// Function to read a string

void readString(char \*str, const char \*prompt) {

printf("%s", prompt);

scanf("%s", str);

}

// Function to perform pattern matching and replacement

void patternMatching(char \*mainStr, const char \*pattern, const char \*replace) {

char result[1000]; // Assuming a fixed size for the result, adjust as needed

int mainLen = strlen(mainStr);

int patLen = strlen(pattern);

int repLen = strlen(replace);

int i, j, k;

for (i = 0; i <= mainLen - patLen; ) {

j = 0;

// Check for pattern match

while (j < patLen && mainStr[i + j] == pattern[j]) {

j++;

}

// If pattern found, replace it and move index accordingly

if (j == patLen) {

for (k = 0; k < repLen; k++) {

result[i + k] = replace[k];

}

i += repLen;

}

else {

result[i] = mainStr[i];

i++;

}

}

// Copy remaining characters from mainStr to result

while (i < mainLen) {

result[i] = mainStr[i];

i++;

}

// Null-terminate the result

result[i] = '\0';

// Copy result back to mainStr

strcpy(mainStr, result);

}

int main() {

char mainStr[1000], pattern[100], replace[100];

// Read main string

readString(mainStr, "Enter the main string: ");

// Read pattern string

readString(pattern, "Enter the pattern string: ");

// Read replace string

readString(replace, "Enter the replace string: ");

// Perform pattern matching and replacement

patternMatching(mainStr, pattern, replace);

// Display the result

printf("Result after pattern matching and replacement: %s\n", mainStr);

return 0;

}