**2.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)**

#include <stdio.h>

#include <string.h>

void replaceString(char \*str, const char \*pattern, const char \*replace) {

char result[1000]; // Assuming a maximum length for the result string

int i, j, k, m, found;

int strLength = strlen(str);

int patLength = strlen(pattern);

int repLength = strlen(replace);

for (i = 0; i < strLength; ) {

found = 0;

// Check if the pattern is present at the current position

for (j = 0; j < patLength; j++) {

if (str[i + j] != pattern[j]) {

found = 0;

break;

}

found = 1;

}

// If the pattern is found, replace it with the replace string

if (found) {

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

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

}

i += repLength;

} else {

result[i] = str[i];

i++;

}

}

result[i] = '\0'; // Add null terminator to the result string

// Copy the modified result back to the original string

for (m = 0; m <= i; m++) {

str[m] = result[m];

}

}

int main() {

char str[1000], pat[100], rep[100];

// Input main string

printf("Enter the main string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0'; // Remove newline character if present

// Input pattern string

printf("Enter the pattern string: ");

fgets(pat, sizeof(pat), stdin);

pat[strcspn(pat, "\n")] = '\0'; // Remove newline character if present

// Input replace string

printf("Enter the replace string: ");

fgets(rep, sizeof(rep), stdin);

rep[strcspn(rep, "\n")] = '\0'; // Remove newline character if present

// Call the function to replace the pattern with the replace string

replaceString(str, pat, rep);

// Output the modified string

printf("Modified string: %s\n", str);

return 0;

}