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)

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>

void readString(char* s, int maxLen, const char* prompt) {
    printf("%s", prompt);
    fgets(s, maxLen, stdin);
    // Removing the trailing newline character
    int i = 0;
    while (s[i] != '\n' && s[i] != '\0') i++;
    s[i] = '\0';
}

int findPattern(const char* str, const char* pat, int start) {
    int i, j;
    for (i = start; str[i] != '\0'; i++) {
        for (j = 0; pat[j] != '\0' && str[i + j] == pat[j]; j++);
        if (pat[j] == '\0')
            return i;
    }
    return -1;
}
```

```
void replacePattern(char* str, const char* pat, const char* rep) {
    char temp[1024];
    int i = 0, j, k;
    int found = 0;
    while (str[i] != '\0') {
        int pos = findPattern(str, pat, i);
        if (pos != -1) {
            found = 1;
            for (j = 0; j < pos; j++) temp[j] = str[j];
            for (k = 0; rep[k] != '\0'; k++, j++) temp[j] = rep[k];
            i = pos;
            while (str[i] == pat[i - pos]) i++;
            while (str[i] != '\0') temp[j++] = str[i++];
            temp[j] = '\0';
            for (i = 0; temp[i] != '\0'; i++) str[i] = temp[i];
            str[i] = '\0';
            i = pos + k;
        } else break;
    }
    if (!found) {
        printf("Pattern not found in the string.\n");
    }
}
```

```
int main() {
    char STR[1024], PAT[100], REP[100];
    readString(STR, 1024, "Enter the main string (STR): ");
    readString(PAT, 100, "Enter the pattern string (PAT): ");
    readString(REP, 100, "Enter the replace string (REP): ");
    replacePattern(STR, PAT, REP);
    printf("Modified String: %s\n", STR);
    return 0;
}
```