```
/* DS Lab 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.
*/

//a. Read a main String (STR), a Pattern String
(PAT) and a Replace String (REP)
#include <stdio.h>
#include <string.h>

int search(char p[], char t[])
{
    int n, m, i, j;
    n = strlen(t);
    m = strlen(p);


    for (i = 0; i <= n - m; i++)
    {
        j = 0;
        while (j < m && p[j] == t[i + j])
            j++;

        if (j == m) return i;
    }

    return -1;
}
```

```c
void main()
{
    char t[50], p[50];
    int pos;

    printf("Enter the text string: ");
    //gets(t);
    //scanf("%s",t);
    scanf("%[^\n]",t);

    printf("Enter the pattern string: ");
    //gets(p);
    scanf("%s",p);
    //scanf("%[^\n]",p);

    pos = search(p, t);
    if (pos == -1)
        printf("Pattern string not found\n");
    else
        printf("Pattern string found at %d\n", pos);
}

/*
Output :
Enter the text string: FUN - UNCLE
Enter the pattern string: UNCLE
Pattern string found at 6

Enter the text string: FUN - UNCLE
Enter the pattern string: ANCLE
Pattern string not found

*/
```

```c
/*
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
*/

#include <stdio.h>
#include <string.h>

int search(char p[], char t[])
{
    int n, m, i, j;

    n = strlen(t);
    m = strlen(p);

    for (i = 0; i <= n - m; i++)
    {
        j = 0;
        while (j < m && p[j] == t[i + j])
            j++;
        if (j == m) return i;
    }

    return -1;
}

void replace(char p[], char t[], char r[], int pos)
{
    int i, k = 0;
    char d[50];


    // Copy the part before the pattern
    for (i = 0; i < pos; i++)
    {
```

```
        d[k] = t[i];
        k++;
    }

    // Copy the replacement string
    for (i = 0; i < strlen(r); i++)
    {
        d[k] = r[i];
        k++;
    }

    // Copy the part after the pattern
    for (i = pos + strlen(p); i <= strlen(t); i++)
    {
        d[k] = t[i];
        k++;
    }

    // Copy the result back to the original string
    for (i = 0; i < strlen(d); i++)
        t[i] = d[i];

}

void main()
{
    char t[50], p[50], r[50];
    int pos;

    printf("Enter the text string: ");
    scanf(" %[^\n]", t);

    printf("Enter the pattern string: ");
    scanf("%s", p);

    printf("Enter the replace string: ");
    scanf("%s", r);
```

```
    pos = search(p, t);
    if (pos == -1)
        printf("Pattern string not found\n");
    else
        printf("Pattern string found at %d\n", pos);

    while (pos != -1)
    {
        replace(p, t, r, pos);
        pos = search(p, t);
    }

    printf("FINAL: %s\n", t);
}

/*
Output :

Enter the text string: TO BE OR NOT TO BE
Enter the pattern string: BE
Enter the replace string: BANG
Pattern string found at 3
FINAL: TO BANG OR NOT TO BANG

Enter the text string: TO BE OR NOT TO BE
Enter the pattern string: Be
Enter the replace string: BANG
Pattern string not found
FINAL: TO BE OR NOT TO BE



*/
```