

Experiment No.-10

Student Name: SANSKAR AGRAWAL

UID: 20BCS5914

Branch: C.S.E.

Section/Group: 20BCS_MM_806B

Semester: 5th

Subject Code: 20CSP-312

Subject Name: Design and Analysis of Algorithm Lab

Aim:

Code and analyze to find all occurrences of a pattern P in a given string S.

Algorithm:

1. $n \leftarrow \text{length}[T]$
2. $m \leftarrow \text{length}[P]$
3. $\Pi \leftarrow \text{COMPUTE-PREFIX-FUNCTION}(P)$
4. $q \leftarrow 0$ // numbers of characters matched
5. for $i \leftarrow 1$ to n // scan S from left to right
6. do while $q > 0$ and $P[q + 1] \neq T[i]$
7. do $q \leftarrow \Pi[q]$ // next character does not match
8. If $P[q + 1] = T[i]$
9. then $q \leftarrow q + 1$ // next character matches
10. If $q = m$ // is all of p matched?
11. then print "Pattern occurs with shift" $i - m$
12. $q \leftarrow \Pi[q]$ // look for the next match

Code:

```
#include <bits/stdc++.h> void computeLPSArray(char* pat,
int M, int* lps); void KMPSearch(char* pat, char* txt)
{
int M = strlen(pat); int N =
strlen(txt); int lps[M];
computeLPSArray(pat, M, lps);
int i = 0; int j = 0;
while ((N - i) >= (M - j)) {

if (pat[j] == txt[i]) { j++; i++;
```

```
}  
if (j == M) { printf("Found pattern at index  
%d ", i - j);  
  
j = lps[j -  
1];  
} else if (i < N && pat[j] !=  
txt[i]) {  
if (j != 0) j =  
lps[j - 1]; else  
i = i + 1;  
}  
} }  
void computeLPSArray(char* pat, int M, int* lps)  
{ int len = 0; lps[0] =  
0; int i = 1; while (i <  
M) { if (pat[i] ==  
pat[len]) { len++;  
lps[i] = len; i++; }  
else { if (len != 0) {  
len = lps[len - 1];  
} else  
{ lps[i] =  
0; i++;  
}  
} } int  
main() {  
char txt[] = "Madhur Sharma"; char  
pat[] = "r"; KMPSearch(pat,  
txt); return 0; }
```

```

1 #include <bits/stdc++.h>
2 void computeLPSArray(char* pat, int M, int* lps);
3 void KMPSearch(char* pat, char* txt)
4 {
5     int M = strlen(pat);
6     int N = strlen(txt);
7     int lps[M];
8     computeLPSArray(pat, M, lps);
9     int i = 0;
10    int j = 0;
11    while ((N - i) >= (M - j)) {
12        if (pat[j] == txt[i]) {
13            j++;
14            i++;
15        }
16        if (j == M) {
17            printf("Found pattern at index %d ", i - j);
18            j = lps[j - 1];
19        }
20        else if (i < N && pat[j] != txt[i]) {
21            if (j != 0)
22                j = lps[j - 1];
23            else
24                i = i + 1;
25        }
26    }
27 }
28 void computeLPSArray(char* pat, int M, int* lps)
29 {
30     int len = 0;
31     lps[0] = 0;
32     int i = 1;
33     while (i < M) {
34         if (pat[i] == pat[len]) {
35             len++;
36             lps[i] = len;
37             i++;
38         }
39         else {
40             if (len != 0) {
41                 len = lps[len - 1];
42             }
43             else {
44                 lps[i] = 0;
45                 i++;
46             }
47         }
48     }
49 }
50 }
51 }
52 int main()
53 {
54     char txt[] = "Madhur Sharma";
55     char pat[] = "r";
56     KMPSearch(pat, txt);
57     return 0;
58 }

```

Complexity Analysis:

Time Complexity: $O(N)$

Output:

```

Found pattern at index 5 Found pattern at index 10
...Program finished with exit code 0
Press ENTER to exit console.

```

Learning Outcome:

1. I have learnt about String



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC
GRADE **A+**
ACCREDITED UNIVERSITY

2.I have learnt about Patterns.

3.I have learnt about how to find the occurrences in a pattern.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			