EXPERIMENT – 3.3

Student Name- Ruchika Raj **UID- 20BCS9285**
Branch- CSE **Section/Group- 615/B**
Semester- 5th **Subject Code- 20CSP-312**
Subject Name–DAA Lab

Aim/Overview of the practical:

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

Algorithm:

1. We will first create the LPS array.
2. Initialize two variables - ‘strIdx’ and ‘patIdx’ to iterate over the string and the pattern, respectively.
3. If ‘pat[patIdx]’ equals ‘str[strIdx]’, we will increment both the indexes.
4. When ‘patIdx’ equals the length of the pattern, this means that the pattern is found in the string. Therefore we print the index and set ‘patIdx’ = LPS[patIdx-1].
5. If ‘pat[patIdx]’ is not equal to ‘str[strIdx]’, we update the patIdx witht he last index that matches with ‘str[strIdx]’ using the LPS array.

Code:

```
#include<bits/stdc++.h>
using namespace std;
```

```
vector<int> getLps(string pat){
    int m = pat.size();

    vector<int>lps(m);
    int prev = 0;
    int ind = 1;
    while (ind < m){

        if (pat[ind]==pat[prev]){
            prev++;
            lps[ind]=prev;
            ind++;
        }

        else if (prev==0){
            lps[ind]=0;
            ind++;
        }

        else{
            prev = lps[prev-1];
        }
    }
    return lps;
}

void solve(string str, string pat){
    vector<int> lps = getLps(pat);

    int n = str.size();
    int m = pat.size();
    int patIdx = 0;
    int strIdx = 0;

    while (strIdx < n){

        if (str[strIdx] == pat[patIdx]){
            patIdx++;
            strIdx++;
        }
        if (patIdx == m){
            cout<<strIdx - m<<' ';
            patIdx = lps[patIdx-1];
        }
        else if (strIdx < n){
```

```
            if (str[strIdx]!=pat[patIdx]){
                if (patIdx != 0)
                    patIdx = lps[patIdx-1];
                else
                    strIdx++;
            }
        }
    }
    cout<<endl;
}

int main(){
    string str,pat;
    cout<<"ENTER THE STRING : ";
    cin>>str;
    cout<<"ENTER THE PATTERN : ";
    cin>>pat;
    solve(str, pat);
}
```

Output:

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE  JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\CU\3rd Year\Sem 5\DAA> cd "d:\CU\3rd Year\Sem 5\DAA\Worksheet\" ; if ($?) { g++ exp3_3a.cpp -o exp3_3a } ; if ($?) { .\exp3_3a }
ENTER THE STRING : RUCHIKA
ENTER THE PATTERN : CHI
2
PS D:\CU\3rd Year\Sem 5\DAA\Worksheet>
```

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE  JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\CU\3rd Year\Sem 5\DAA> cd "d:\CU\3rd Year\Sem 5\DAA\Worksheet\" ; if ($?) { g++ exp3_3a.cpp -o exp3_3a } ; if ($?) { .\exp3_3a }
ENTER THE STRING : AABAACAADAABAAABAA
ENTER THE PATTERN : AABA
0 9 13
PS D:\CU\3rd Year\Sem 5\DAA\Worksheet>
```

LEARNING OUTCOMES :

- We learnt about the properties of string.
- We learnt about the KMP algorithm.
- We learnt about the LPS(Longest proper Prefix) array