

# KMP SEARCH ALGO

## CODE:

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;
bool flag=false;
void computeLPSArray(string& pat, int M, vector<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++;
            }
        }
    }
}
```

```

    }
}

vector<int> KMPSearch(string& pat, string& txt)
{
    int M = pat.length();
    int N = txt.length();

    vector<int> lps(M);

    vector<int> result;

    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) {
            flag=true;
            result.push_back(i - j + 1);
            j = lps[j - 1];
        }

        else if (i < N && pat[j] != txt[i]) {

            if (j != 0)
                j = lps[j - 1];
            else
                i = i + 1;
        }
    }

    return result;
}

```

```

int main()
{
    string txt ;
    cout<<"Enter the text: ";
    cin>>txt;
    string pat;
    cout<<"Enter the pattern: ";
    cin>>pat;
    vector<int> result = KMPSearch(pat, txt);
    if(flag==false)cout<<"Sorry pattern does not exist in the string";
    else{
        cout<<"Pattern exists in the string at index: ";
        for (int i = 0; i < result.size(); i++) {
            cout << result[i]-1 << " ";
        }
    }
    return 0;
}

```

## OUTPUT:

```

Enter the text: ababcdababffgghh
Enter the pattern: abab
Pattern exists in the string at index: 0 6

```

```

Enter the text: ababaccd
Enter the pattern: z
Sorry pattern does not exist in the string

```