# Experiment 12

AIM

THEORY

ALGORITHM:

#include<string.h>

#include<iostream>

using namespace std;

void PrefixFunc(char\* pattern, int M, int\* prefix)

{

    int len = 0;

    prefix[0] = 0;

    int i = 1;

    while (i < M) {

        if (pattern[i] == pattern[len]) {

            len++;

            prefix[i] = len;

            i++;

        }

        else

        {

            if (len != 0) {

                len = prefix[len - 1];

            }

            else

            {

                prefix[i] = 0;

                i++;

            }

        }

    }

}

void KMPSearch(char\* pattern, char\* txt)

{

    int M = strlen(pattern);

    int N = strlen(txt);

    int prefix[M];

    PrefixFunc(pattern, M, prefix);

    int i = 0;

    int j = 0;

    while ((N - i) >= (M - j)) {

        if (pattern[j] == txt[i]) {

            j++;

            i++;

        }

        if (j == M) {

            cout<<"Found pattern at index : "<<i - j<<endl;

            j = prefix[j - 1];

        }

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

            if (j != 0)

                j = prefix[j - 1];

            else

                i = i + 1;

        }

    }

}

int main()

{

    char txt[] = "SIDDHARTH BHATIA";

    char pattern[] = "HA";

    cout<<"Text : "<<txt<<endl;

    cout<<"Pattern : "<<pattern<<endl;

    KMPSearch(pattern,txt);

    return 0;

}

OUTPUT:

