**DAA LAB: 8**

**PRATEEK MOHANTY**

**20BCE1482**

**NAIVE STRING MATCHING**

**Q1**

**CODE**

#include <iostream>

using namespace std;

// string matching

int main()

{

    string a, b;

    cin >> a >> b;

    int n = a.length();

    int m = b.length();

    for (int i = 0; i <= n - m; i++)

    {

        for (int j = 0; j < m; j++)

        {

            if (a[i + j] != b[j])

                break;

            if (j == m - 1)

            {

                cout << "Match found at index " << i << endl;

                break;

            }

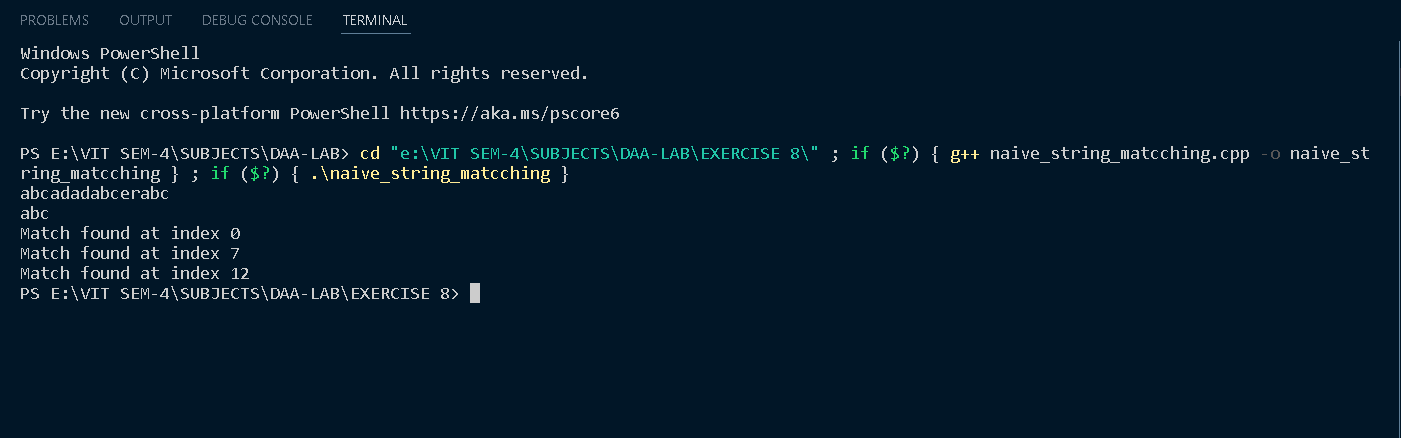
        }

    }

    return 0;

}

**OUTPUT**

****

**KMP ALGORITHM**

**Q2**

**CODE**

#include <iostream>

using namespace std;

void LpsArray(string pattern, int lenPat, int \*lps)

{

    int a = 0;

    // first char is always 0

    lps[0] = 0;

    int k = 1;

    int temp\_length = 0;

    while (k < lenPat)

    {

        if (pattern[k] == pattern[temp\_length])

        {

            temp\_length++;

            lps[k] = temp\_length;

            k++;

        }

        else

        {

            // condition when no match is found

            if (temp\_length != 0)

            {

                temp\_length = lps[temp\_length - 1];

            }

            else

            {

                // if character not present

                lps[k] = 0;

                k++;

            }

        }

    }

}

void kmp\_match(string pattern, string text)

{

    int m = pattern.length();

    int n = text.length();

    int lps[m];

    LpsArray(pattern, m, lps);

    int i = 0;

    int j = 0;

    while (i < n)

    {

        if (pattern[j] == text[i])

        {

            i++;

            j++;

        }

        else if (i < n && pattern[i] != text[j])

        {

            if (j != 0)

            {

                j = lps[j - 1];

            }

            else

            {

                i++;

            }

        }

        if (j == m - 1)

        {

            cout << "Pattern found at index: " << i - j << endl;

            j = lps[j - 1];

        }

    }

}

int main()

{

    string a, b;

    cout << "Enter the first and second string"

         << " ";

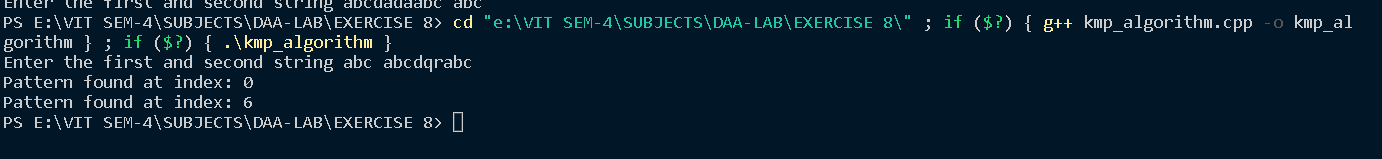
    cin >> a >> b;

    kmp\_match(a, b);

    return 0;

}

**OUTPUT**

****