**LPS DAA EXERCISE 8**

**PRATEEK MOHANTY**

**20BCE1482**

**Q1**

**OPTIMZED NAIVE PATTERN MATCHING**

**CODE**

#include <iostream>

#include <vector>

using namespace std;

// optimzed string matching algorithm

int main()

{

    string a, b;

    cin >> a >> b;

    int n = a.length();

    int m = b.length();

    int k = 0;

    // In KMP String pre processing is required

    int pre[10];

    // vector<int>pre;

    for (int i = 0; i < n; i++)

    {

        if (b[0] == a[i])

        {

            pre[k++] = i;

        }

    }

    //  for(int i=0;i<k;i++){

    //      cout<<pre[i]<<" ";

    //  }

    int e = k;

    int l = 0;

    for (int i = pre[l]; i < n && l < k; i++)

    {

        for (int j = 0; j < m; j++)

        {

            if (a[i + j] != b[j])

                break;

            if (j == m - 1)

            {

                cout << "Match found" << endl;

            }

        }

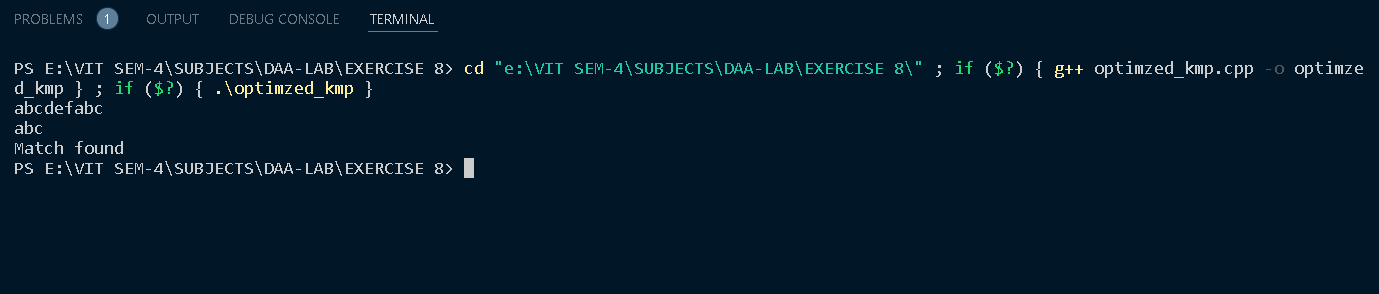
        l++;

    }

    return 0;

}

**OUTPUT**

****

**Q2**

**STRING MATCHING WITH $ SIGN**

**CODE**

#include <iostream>

using namespace std;

int main()

{

    cout << "Enter the pattern and the text" << endl;

    string pattern, text;

    cin >> pattern >> text;

    cout << "Enter the value of the dollar symbol" << endl;

    char e;

    cin >> e;

    for (int i = 0; i < text.length(); i++)

    {

        if (text[i] == '$')

        {

            text[i] = e;

        }

    }

    int k = 0;

    int n = text.length();

    int m = pattern.length();

    // In KMP String pre processing is required

    int pre[10];

    // vector<int>pre;

    for (int i = 0; i < n; i++)

    {

        if (pattern[0] == text[i])

        {

            pre[k++] = i;

        }

    }

    //  for(int i=0;i<k;i++){

    //      cout<<pre[i]<<" ";

    //  }

    int l = 0;

    for (int i = pre[l]; i < n && l < k; i++)

    {

        for (int j = 0; j < m; j++)

        {

            if (text[i + j] != pattern[j])

                break;

            if (j == m - 1)

            {

                cout << "Match found" << endl;

            }

        }

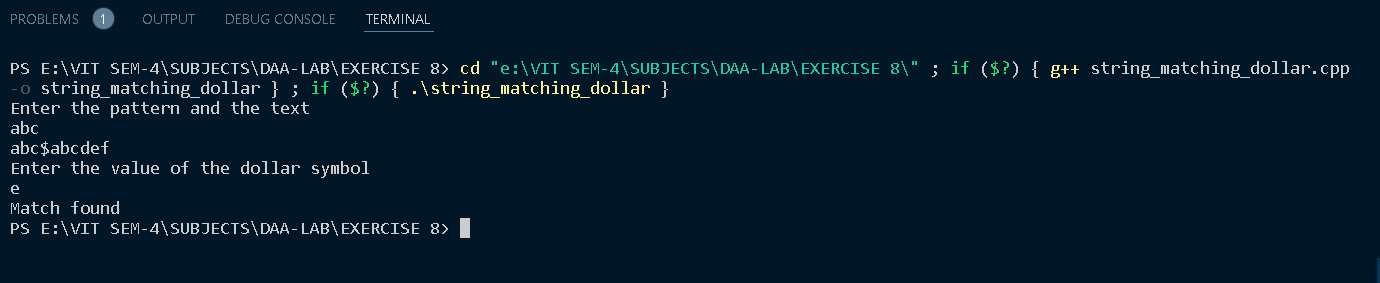
        l++;

    }

    return 0;

}

**OUTPUT**

****

**STRING MATCHING WITH $ SIGN (STRING)**

**Q3**

**CODE**

#include <iostream>

using namespace std;

int searchString(string pattern, string text)

{

    int m = pattern.length();

    int n = text.length();

    int i = 0;

    while (i <= n - m)

    {

        int j;

        for (int j = 0; j < m; j++)

        {

            if (text[i + j] != pattern[j])

                break;

            if (j == m)

            {

                return i;

                i = i + m;

            }

            else if (j == 0)

                i++;

            else

                i = i + j;

        }

        i++;

    }

    return 0;

}

int main()

{

    string txt = "ABCEABCDABCEABCD";

    cout << "check the regex BC$ABCD where $ can be any symbol" << endl;

    string pat1 = "BC";

    string pat2 = "ABCD";

    int i1 = searchString(pat1, txt);

    int i2 = searchString(pat2, txt) + pat2.length() - 1;

    if (i1 < i2)

    {

        cout << "Pattern found at index " << i1 << " to index "

             << "i2";

        cout << " Value of $ will be"

             << " ";

        string a;

        for (int i = i1 + 1; i < i2 - 1; i++)

        {

            a += txt[i];

        }

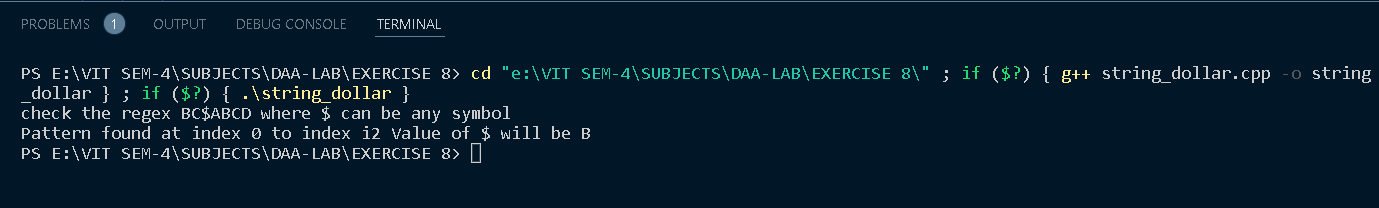
        cout << a;

    }

    return 0;

}

**OUTPUT**

****