**EXPERIMENT N0-3**

AIM: Write a program to implement the KMP algorithm.

**CODE:**

#include <stdio.h>

#include <sptring.h>

char txt[100], pat[100];

int M, N, lps[100], j = 0, i = 0;

void computeArray()

{

int len = 0, i;

lps[0] = 0;

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++;

}

}

}

}

void KMPSearch()

{

int j = 0, i = 0;

M = strlen(pat);

N = strlen(txt);

computeArray();

while (i < N)

{

if (pat[j] == txt[i])

{

j++;

i++;

}

if (j == M)

{

printf("Found pattern at index %d \n", i - j);

j = lps[j - 1];

}

else if (pat[j] != txt[i])

{

if (j != 0)

j = lps[j - 1];

else

i = i + 1;

}

}

}

int main()

{

printf("\n ENTER THE TEXT: ");

gets(txt);

printf("\n ENTER THE PATTERN : ");

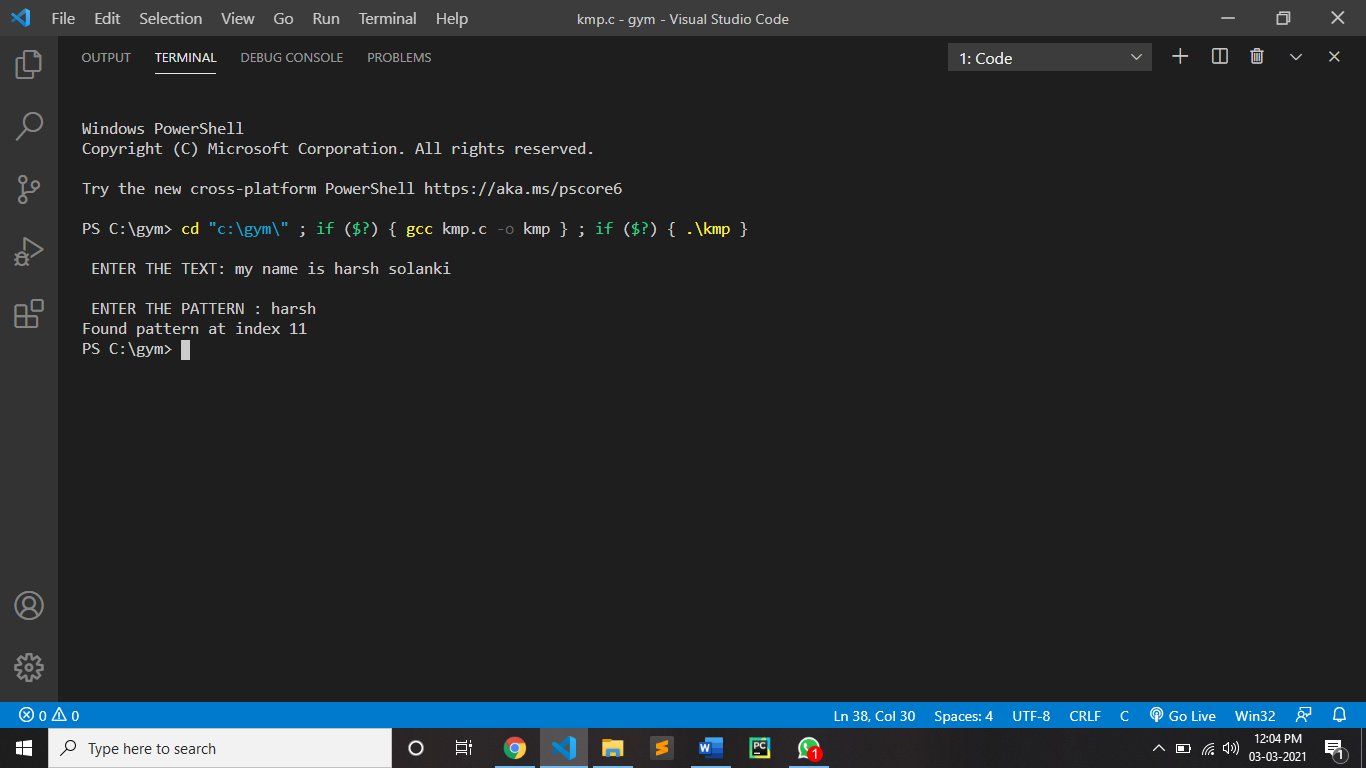
gets(pat);

KMPSearch();

return 0;

}

**OUTPUT:**

****

**CONCLUSION:**

By performing Knuth Morris Pratt algorithm we can conclude that The time complexity of KMP algorithm apart from Suffix Array Calculation is O(m) and Suffix generation array takes O(n). So the total time complexity of KMP algorithm is O(m+n). Here m= size of array and n= size of pattern to be searched. Worst case for KMP algorithm is O(n). and the space complexity is O (m)