|  |  |
| --- | --- |
| **Name** | Manish Shashikant Jadhav |
| **UID** | 2023301005 |
| **Subject** | Design and Analysis of Algorithms (DAA) |
| **Experiment No.** | 10 |
| **Aim** | To implement string matching using Knuth-Morris-Pratt algorithm. |
| **Code:** | #include <stdio.h>  #include <string.h>  void computeLPSArray(char \**pat*, int *M*, 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++;              }          }      }  }  void KMPSearch(char \**pat*, char \**txt*) {      int M = strlen(*pat*);      int N = strlen(*txt*);      int lps[M];      computeLPSArray(*pat*, M, lps);      int i = 0; *// index for txt[]*      int j = 0; *// index for pat[]*      while (i < N) {          if (*pat*[j] == *txt*[i]) {              j++;              i++;          }          if (j == M) {              printf("Pattern found at index %d\n", i - j);              j = lps[j - 1];          } else if (i < N && *pat*[j] != *txt*[i]) {              if (j != 0)                  j = lps[j - 1];              else                  i = i + 1;          }      }  }  int main() {      char txt[100], pat[100];      printf("Enter the text: ");      fgets(txt, sizeof(txt), stdin);      txt[strcspn(txt, "\n")] = '\0'; *// Remove newline character*      printf("Enter the pattern to search: ");      fgets(pat, sizeof(pat), stdin);      pat[strcspn(pat, "\n")] = '\0'; *// Remove newline character*      printf("Text: %s\n", txt);      printf("Pattern: %s\n", pat);      KMPSearch(pat, txt);      return 0;  } |
| **Output** |  |
| **Conclusion** | Hence, by completing this experiment I came to know about implementation of string matching using Knuth-Morris-Pratt algorithm. |