KMP

```
#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;
    int j = 0;
    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[1000], pat[1000];
    printf("Enter the text: ");
    fgets(txt, sizeof(txt), stdin);
    printf("Enter the pattern: ");
    fgets(pat, sizeof(pat), stdin);
    txt[strcspn(txt, "\n")] = '\0';
    pat[strcspn(pat, "\n")] = '\0';
    KMPSearch(pat, txt);
    return 0;
}
```

OUTPUT