## Design and Analysis of Algorithms – 20ISL57A

## Program 12 - Implement and compare Simple string matching and KMP string matching algorithm.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<string.h>
int lps[100];
void computeLPSArray(char pattern[])
{
	int len=0, i=1, m, n;
	lps[0] = 0;
	m= strlen(pattern);
	while(i< m)
	 {
		if(pattern[i] == pattern[len])
		{
			len++;
			lps[i] = len;
			i++;
		 }
		else
		{
			if(len != 0 )
				len = lps[len-1];
			else
			{
				lps[i] = 0;
				i++;
			 }
		 }
	}
}
void kmp(char text[],char pattern[])
{
	int j=0,i=0,m,n;
```

```
        m= strlen(pattern);
         n = strlen(text);
        computeLPSArray(pattern);
        while(i< n)
         {
                if(pattern[j] == text[i])
                {
                        j++;
                        i++;
                 }
                if (j == m)
                {
                        printf("Using KMP pattern found at index %d \n", i-j);
                        j = lps[j-1];
                }
                else if(pattern[j] != text[i])
                {
                        if(j != 0)
                                j = lps[j-1];
                        else
                                i = i+1;
                 }
         }
}
void bruteforce(char text[],char pattern[])
{
                int i,j,k,m,n,flag=1;
                n=strlen(text);
                m=strlen(pattern);
                for(i=0;i<=n-m;i++)
                {
                        j=0;
                        while(j<m && pattern[j]==text[j+i])
                        {
                                j++;
                                if(j==m)
                                {
```

```
                                        flag=1;
                                        k=i;
                                }
                                else
                                        flag=0;
                        }
          }
          if (flag==1)
                printf("Using Bruteforce pattern found at index %d\n",k);
        else
                printf("Using Bruteforce: No Match found");
}
int main()
{
        char text[50],pattern[50];
        clock_t st1,st2,end1,end2;
        printf("Enter the text\n");
        gets(text);
        printf("Enter the pattern\n");
        gets(pattern);
        st1=clock();
        bruteforce(text,pattern);
        end1=clock();
        printf("Time required for bruteforce match %f\n",(float)(end1-st1)/CLOCKS_PER_SEC);
        st2=clock();
        kmp(text,pattern);
        end2=clock();
        printf("Time required for kmp match %f\n",(float)(end2-st2)/CLOCKS_PER_SEC);
}
```