Q9. a) Implement KMP Algorithm. Analyze its time complexity.

TIME COMPLEXITY: O(m + n)

SPACE COMPLEXITY: O(n ^ 2)

```
#include <iostream>
#include <queue>
#include <vector>
#include <string>
#include <map>
#include <cmath>

using namespace std;

void computeLPSArray(string & pat, int M, vector<int> & lps);


void KMPSearch(string & pat, string & txt)
{
    int M = int(pat.size());
    int N = int(txt.size());
    vector<int> lps(M);
    computeLPSArray(pat, M, lps);
    int i = 0, j = 0;
    while (i < N)
    {
        if (pat[j] == txt[i])
        {
            j++;
            i++;
        }
        if (j == M)
        {
            cout << "Found pattern at index " <<  i - j << endl;
            j = lps[j - 1];
        }
        else if (i < N && pat[j] != txt[i])
        {
            if (j != 0)
                j = lps[j - 1];
            else
                i = i + 1;
        }
    }
}

void computeLPSArray(string & pat, int M, vector<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)
                len = lps[len - 1];
            else
            {
                lps[i] = 0;
                i++;
            }
        }
    }
}


int main()
{
    string txt = "ABABDABACDABABCABAB";
    string pat = "ABABCABAB";
    cout << "TEXT: " << txt << endl;
    cout << "PATTERN: " << txt << endl;
    KMPSearch(pat, txt);
    return 0;
}
```

OUTPUT:

```
TEXT: ABABDABACDABABCABAB
PATTERN: ABABDABACDABABCABAB
Found pattern at index 10
Program ended with exit code: 0
```