```
let txt = "abcab";
let pattern = "ab";

function searchPatternKMP(txt,pattern){
    let n = txt.length;
    let m = pattern.length;
    let lps = new Array(m).fill(0);

    // compute lps (Longest prefix suffix)
    computeLPS(pattern,m,lps);
    // search pattern KMP method
    let i = 0;
    let j = 0;
    let result = [];
    while(i<n){
        if(txt[i] == pattern[j]){
            i++;
            j++;
        }
        // found
        if(j == m){
            result.push(i - j);
            j = lps[j - 1];
        }else if(txt[i] != pattern[j]){
            if(j!=0){
                j = lps[j-1];
            }else{
                i++;
            }
        }
    }
    return result;

}

function computeLPS(pattern,m,lps){
    let length = 0;
    let i = 1;
    lps[0] = 0;
    while(i<m){
        if(pattern[i] == pattern[length]){
            length++;
            lps[i] = length;
            i++;
        }else{
            if(length != 0){
                length = lps[length - 1];
            }else {
                lps[i] = 0;
                i++;
            }
        }
    }

}

console.log(searchPatternKMP(txt,pattern));
```