Rabin Karp

```
let a = 'baaaabde'
let b = 'aab'
let c = 'ace'
let d = 'aba'

function calculateHash(str){

    let sum = 0
    for(let i = 0; i<str.length; i++){
      sum +=  str.charCodeAt(i) * Math.pow(10,str.length-1-i)

    }
    return sum
}


function patternCheck(a,b){
for(let i = 0; i<a.length; i++){
    if(a[i]!==b[i]){

        return false
    }

}
return true
}
function rabinKarp(str,sbstr){
    let sbstrHash = calculateHash(sbstr);
    let i = 0
    while( i<=  str.length- sbstr.length ){
        let currentSlice = str.slice(i, i + sbstr.length);

        if(sbstrHash === calculateHash(currentSlice)){
           if(patternCheck(currentSlice, sbstr)) return true
        }
        i++
    }
    return false
}


console.log(rabinKarp(a,b))
console.log(rabinKarp(a,c))
console.log(rabinKarp(a,d))
```

KMP Algorithm

```
//KMP Algo
//Calculate the indices of the occurrences of strings s in it

let string = "abcabcd";
let substring = "abcd";
let string2 = "abcabcdfgh";
let substring21 = "fgh";
let substring22 = "cfgh";

const bruteForceMatch = (string, substring) => {
    //O(n*m)
    if (substring === "") return 0;
    for (let i = 0; i <= string.length - substring.length; i++) {
        let j = 0;
        for (; j < substring.length; j++) {
            if (substring[j] !== string[i + j]) break;
        }
        if (j === substring.length) return i;
    }
    return -1;
};

console.log(bruteForceMatch(string, substring))
let substring2 = "abca";
let substring3 = "abda";
function buildPrefixTable(s) {
    let table = [0];
    let i = 1;
    let j = 0;
    while (i < s.length) {
        if (s[i] === s[j]) {
            j++;
            table[i] = j;
            i++;
        } else if (j > 0) {
            //do not match
            j = table[j - 1];
        } else {
            //do not match j === 0
            table[i] = 0;
            i++;
        }
    }
    return table;
}

function checkCommon(string, substring) {
    let s1 = string.split("");
    let table = buildPrefixTable(substring);
    let sb = substring.split("");

    let i = 0,
```

```
        j = 0,
        x = 0;
    let flag = false;
    while (i < s1.length) {
        if (s1[i] === sb[j]) {
            if (j === sb.length - 1 && s1[i] === sb[j]) {
                flag = true;
                break;
            }

            i++;
            j++;
        } else if (j === 0) {
            i++;
        } else {
            j = table[j];
        }
    }
    return flag;
}

console.log(checkCommon(string, substring));
console.log(checkCommon(string, substring2));
console.log(checkCommon(string, substring3));
console.log(checkCommon(string2, substring21));
console.log(checkCommon(string2, substring22));
```