

```

import java.util.Arrays;

public class KMP {

public static void KMPStringMultipleTimes (String text, String pattern) {
    boolean found = false;
    int lps[] = computeTempString(pattern);
    int j = 0, i = 0;
    while (i < text.length()) {
        if (text.charAt(i) == pattern.charAt(j)) {
            ++i; ++j;
        } else {
            if (j != 0) {
                j = lps[j - 1];
            } else {
                ++i;
            }
        }
        if (j == pattern.length()) {
            System.out.println("found match at : " + (i - pattern.length()));
            j = lps[j - 1];
            found = true;
        }
    }
    if (!found) System.out.println("not found");
}

public static void KMPChar (char [] text, char [] pattern) {
    int lps[] = computeTempChar(pattern);
    int j = 0, i = 0;
    boolean found = false;
    while (i < text.length && j < pattern.length) {
        if (text[i] == pattern[j]) {
            ++i; ++j;
        } else {
            if (j != 0) {
                j = lps[j - 1];
            } else {
                ++i;
            }
        }
        if (j == pattern.length) {
            System.out.println("found match at : " + (i - pattern.length));
            j = lps[j - 1];
            found = true;
        }
    }
    if (!found) {
        System.out.println("Not found");
    }
}

public static int [] computeTempChar (char pattern[]) {
    int lps[] = new int[pattern.length];
    int index = 0;
    for (int i = 1; i < pattern.length; ) {
        if (pattern[i] == pattern[index]) {
            lps[i] = index + 1;
            ++index; ++i;
        } else {
            if (index != 0) {
                index = lps[index - 1];
            } else {

```

```

        lps[i] = 0;
        ++i;
    }
}

return lps;
}

public static int[] computeTempString(String pattern) {
    int lps [] = new int [pattern.length()];
    int index = 0;
    for (int i = 1; i < pattern.length(); i) {
        if (pattern.charAt(i) == pattern.charAt(index)) {
            lps[i] = index + 1;
            ++index; ++i;
        } else { if (index != 0) {
            index = lps[index - 1];
        } else {
            lps[i] = 0;
            ++i;
        }
    }
}

return lps;
}

public static void main(String[] args) {
    String a = "This is a testy test text";
    KMPStringMultipleTimes(a, "test");
}
}

```