

<b>Name</b>	<b>Harsh Chandra</b>
<b>UID no.</b>	<b>2021700013</b>
<b>Experiment No.</b>	<b>10</b>

<b>AIM:</b>	<b>String Matching algorithms (To implement Robin Karp algorithm)</b>
-------------	---

<b>Program 1</b>
------------------

<b>ALGORITHM/ THEORY:</b>	<p>Rabin Karp algorithm is based on hashing technique. It first computes the hash value of p and <math>t_s</math>.</p> <ul style="list-style-type: none"> <li>• If hash values are same, i.e. if <math>\text{hash}(p) = \text{hash}(t_s)</math>, we check the equality of inverse hash similar to a naïve method. If hash values are not same, no need to compare actual string.</li> <li>• On the hash match, actual characters of both strings are compared using brute force approach. If the pattern is found, then it is called <i>hit</i>. Otherwise, it is called a <i>spurious hit</i>.</li> </ul>
-------------------------------	--

<b>PROGRAM:</b>	<pre> #include &lt;stdio.h&gt; #include &lt;string.h&gt;  #define d 256  void search(char pat[], char txt[], int q) {     int M = strlen(pat);     int N = strlen(txt);     int i, j;     int p = 0; // hash value for pattern     int t = 0; // hash value for txt     int h = 1;      for (i = 0; i &lt; M - 1; i++)         h = (h * d) % q; </pre>
-----------------	--

```

    for (i = 0; i < M; i++) {
        p = (d * p + pat[i]) % q;
        t = (d * t + txt[i]) % q;
    }

    for (i = 0; i <= N - M; i++) {

        if (p == t) {
            for (j = 0; j < M; j++) {
                if (txt[i + j] != pat[j])
                    break;
            }

            // if p == t and pat[0...M-1] = txt[i, i+1,
            // ...i+M-1]
            if (j == M)
                printf("Pattern found at index %d \n", i);
        }

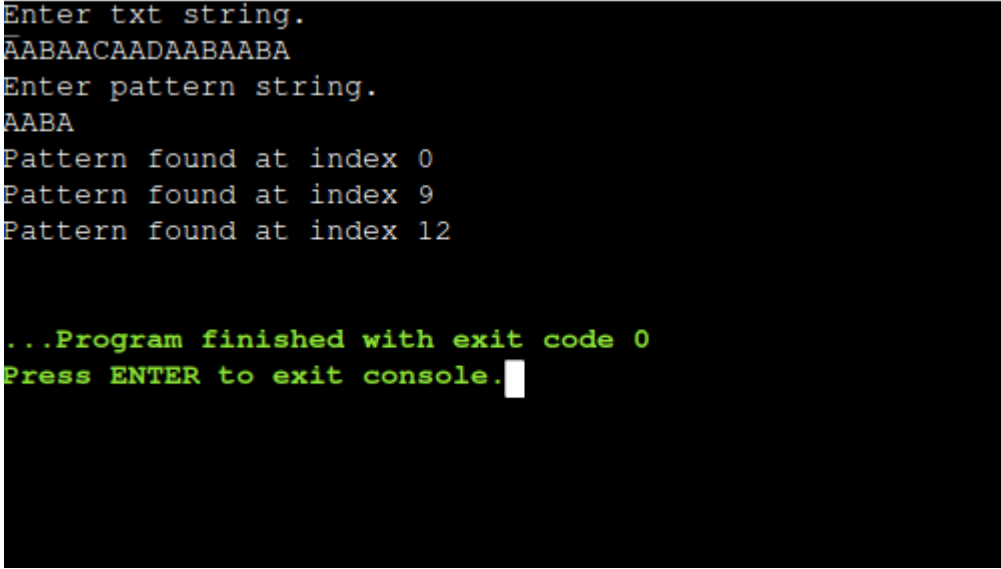
        // Calculate hash value for next window of text:
        // Remove leading digit, add trailing digit
        if (i < N - M) {
            t = (d * (t - txt[i] * h) + txt[i + M]) % q;

            if (t < 0)
                t = (t + q);
        }
    }
}

int main()
{
    char txt[50];
    char pat[50];
    printf("Enter txt string.\n");
    scanf("%s", txt);

    printf("Enter pattern string.\n");
    scanf("%s", pat);

```

	<pre> int q = 101; search(pat, txt, q); return 0; } </pre>
<b>RESULT:</b>	 <pre> Enter txt string. AABAACAADAABAABA Enter pattern string. AABA Pattern found at index 0 Pattern found at index 9 Pattern found at index 12  ...Program finished with exit code 0 Press ENTER to exit console. </pre>
<b>CONCLUSION:</b>	<p>From this experiment, I understood how to implement string matching algorithm Rabin Karp algorithm to search pattern in a string.</p>