

Name	Siddhesh Sonar
UID no.	2021700063
Experiment No.	10

AIM:	To implement Robin Karp algorithm
Program	
PROBLEM STATEMENT :	To implement Robin Karp algorithm
ALGORITHM/ THEORY:	<p>The Rabin-Karp string matching algorithm calculates a hash value for the pattern, as well as for each M-character subsequences of text to be compared. If the hash values are unequal, the algorithm will determine the hash value for next M-character sequence. If the hash values are equal, the algorithm will analyze the pattern and the M-character sequence. In this way, there is only one comparison per text subsequence, and character matching is only required when the hash values match.</p> <p>Algorithm:</p> <p>RABIN-KARP-MATCHER (T, P, d, q)</p> <ol style="list-style-type: none"> 1. $n \leftarrow \text{length}[T]$ 2. $m \leftarrow \text{length}[P]$ 3. $h \leftarrow d^{m-1} \bmod q$ 4. $p \leftarrow 0$ 5. $t0 \leftarrow 0$ 6. for $i \leftarrow 1$ to m 7. do $p \leftarrow (dp + P[i]) \bmod q$ 8. $t0 \leftarrow (dt0 + T[i]) \bmod q$ 9. for $s \leftarrow 0$ to $n-m$ 10. do if $p = ts$

11. then if $P[1.....m] = T[s+1.....s + m]$
12. then "Pattern occurs with shift" s
13. If $s < n-m$
14. then $ts+1 \leftarrow (d(ts-T[s+1]h)+T[s+m+1])\text{mod } q$

PROGRAM:

```
#include <stdio.h>
#include <string.h>

void search(char pat[], char txt[], int q)
{
    int M = strlen(pat);
    int N = strlen(txt);
    int p = 0, t = 0, h = 1, d = 256;
    int i, j;

    for (i = 0; i < M - 1; i++) {
        h = (h * d) % q;
    }

    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 (j == M) {
                printf("\nPattern found at index %d \n", i);
            }
        }

        if (i < N - M) {
            t = (d * (t - txt[i] * h) + txt[i + M]) % q;

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

```

    }
}

int main()
{
    char txt[100], pat[100];
    printf("\nEnter the text: ");
    scanf("%[^\\n]s",txt);
    getchar();
    printf("\nEnter the pattern: ");
    scanf("%[^\\n]s",pat);
    int q = 101;
    search(pat, txt, q);
    return 0;
}

```

RESULT:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... Code + v [] [] ... ^ x

c:\Siddhesh\Github\DAA\DAA_Exp_10>cd "c:\Siddhesh\Github\DAA\DAA_Exp_10\" &
& gcc Rabin_Karp.c -o Rabin_Karp && "c:\Siddhesh\Github\DAA\DAA_Exp_10\Rab
in_Karp

Enter the text: My Name is Siddhesh

Enter the pattern: Siddhesh

Pattern found at index 11

c:\Siddhesh\Github\DAA\DAA_Exp_10>

```

CONCLUSION:

Successfully understood Rabin Karp algorithm and implemented it in C program to match strings.

