NAME:	Shubham Vishwakarma
UID No.	2021700071
BRANCH:	S.Y CSE-DS
BATCH:	D
SUBJECT	Design and Analysis of Algorithms
EXPERIMENT No.	10
Date of Performance	16/04/2023
Date of Submission	20/04/2023

AIM:	**String Matching algorithms (To implement Robin Karp algorithm)**
Program 1	
PROBLEM STATEMENT :	Implement the Robin Karp algorithm to the given sequence.

ALGORITHM/ THEORY:	Start
	  pat_len := pattern Length
	  str_len := string Length
	  patHash := 0 and strHash := 0, h := 1
	  maxChar := total number of characters in character set
	for index i of all character in the pattern, do
	  h := (h*maxChar) mod prime
	for all character index i of pattern, do
	  patHash := (maxChar*patHash + pattern[i]) mod prime
	  strHash := (maxChar*strHash + text[i]) mod prime
	for i := 0 to (str_len - pat_len), do
	  if patHash = strHash, then
	    for charIndex := 0 to pat_len -1, do
	      if text[i+charIndex] ≠ pattern[charIndex], then
	      break
	if charIndex = pat_len, then
	  print the location i as pattern found at i position.
	if i < (str_len - pat_len), then
	  strHash := (maxChar*(strHash – text[i]*h)+text[i+patLen]) mod prime, then
	  if strHash < 0, then
	  strHash := strHash + primeEnd

PROGRAM:	
| | ```c
#include <stdio.h>
#include <string.h>
int main()
{
    char txt[80], pat[80];
    int q;
    printf ("Enter the container string\n");
    scanf ("%s", &txt);
    printf ("Enter the pattern to be searched\n");
    scanf ("%s", &pat);
    int d = 256;
    printf ("Enter a prime number : ");
    scanf ("%d", &q);
    int M = strlen (pat);
    int N = strlen (txt);
    int i, j;
    int p = 0;
    int t = 0;
    int h = 1;
    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", i);
        }
        if (i < N - M)
        {
            t = (d * (t - txt[i] * h) + txt[i + M]) % q;
            if (t < 0)
            t = (t + q);
        }
    }
    return 0;
}
``` |

RESULT:

```
PS C:\Users\smsha\Desktop\SEM 4\DAA\Practicals\Exp10\output> & .\'pattern.exe'
Enter the container string
ASGHJSAKASJAAJKDS
Enter the pattern to be searched
J
Enter a prime number : 2

Pattern found at index 4
Pattern found at index 10
Pattern found at index 13
PS C:\Users\smsha\Desktop\SEM 4\DAA\Practicals\Exp10\output>
```

CONCLUSION :	Rabin-Karp algorithm **can be used to check image similarity based on hash value calculation**. The way the calculation works is the same as that done to string matching. Hence we achieved the aim of the experiment.