Bharatiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.

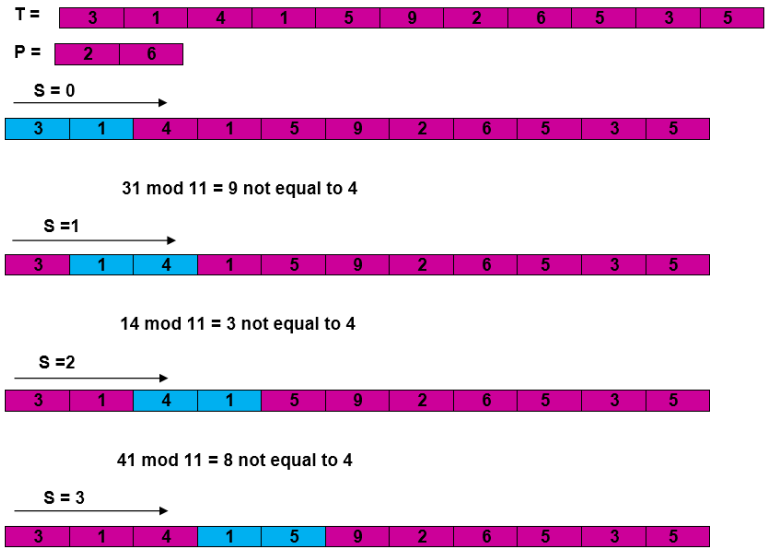
Experiment No. 10

Aim – To implement String Matching Algorithm

Details – The Rabin-Karp string matching algorithm calculates a hash value for the pattern, as well as for each M-character sub-strings of text to be compared. If the hash values are unequal, the algorithm will determine the hash value for next M-character string. If the hash values are equal, the algorithm will analyze the pattern and the M-character string. In this way, there is only one comparison per text string, and character matching is only required when the hash values match.

```
RABIN-KARP-MATCHER(T, P, d, q)
 1  n ← length[T]
 2  m ← length[P]
 3  h ← d^(m−1) mod q
 4  p ← 0
 5  t0 ← 0
 6  for i ← 1 to m                    ▷ Preprocessing.
 7       do p ← (dp + P[i]) mod q
 8          t0 ← (dt0 + T[i]) mod q
 9  for s ← 0 to n − m                ▷ Matching.
10       do if p = ts
11             then if P[1..m] = T[s + 1..s + m]
12                       then print "Pattern occurs with shift" s
13          if s < n − m
14             then ts+1 ← (d(ts − T[s + 1]h) + T[s + m + 1]) mod q
```

Example: For string matching, working module q = 11, how many spurious hits does the Rabin-Karp matcher encounters in Text T = 31415926535



Some Links:

1. YouTube Video: https://www.youtube.com/watch?v=qQ8vS2btsxI&t=169s
2. Reading Resource: https://www.javatpoint.com/daa-rabin-karp-algorithm

---

Input – Text and Pattern strings
Output – Start and end Indices of Text string where there is match of pattern
Submission –

1) C/C++ source code of implementation
2) Verified output for the written source code with multiple inputs
3) One page report of Exp. 10