* String Matching Algo.

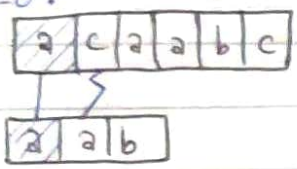
- Need - finding all ~~occure~~ occurrences of a pattern in the text.
- Appl^{n}s: patterns in DNA, internet SEs.
- text array $T[1.....n]$ & pattern is an array $P[1....m]$

1.) Naive Approach

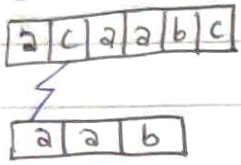
Finds all valid shifts using a loop that checks the cond^{n}

$P[1....m] = T[s+1.....s+m]$

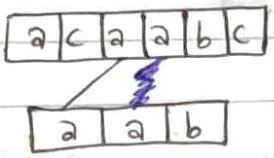
s=0:



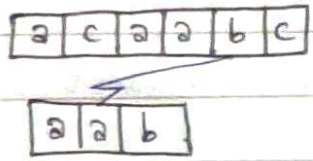
s=1:



s=2:



s=3:



Pattern occurs with shift 2

- Naive-String-Matcher (T, P)

```
n = T.length
m = P.length
for s = 0 to n-m
    if P[1....m] == T[s+1..... s+m]
        print ("Pattern occurs with shift 's'")
```

Time: $O((n-m+1)\,m)$

2.) Rabin-Karp Algo.

Text T :

3	1	4	1	5	9	2	6	5	3	5

Pattern P:

2	6

Choose a random prime no., $q = 11$

Let $p = P \bmod q$

$= 26 \bmod 11 = 4$

Let t_s denote modulo q for text of length m

$t_{s_0} = 31 \bmod 11 = 9$

$t_{s_1} = 14 \bmod 11 = 3$

$t_{s_2} = 41 \bmod 11 = 8$

$t_{s_3} = 15 \bmod 11 = 4$ ⟵ Spurious hit

$t_{s_4} = 59 \bmod 11 = 4$

$t_{s_5} = 92 \bmod 11 = 4$

$t_{s_6} = 26 \bmod 11 = 4$ ⟵ valid match

$t_{s_7} = 65 \bmod 11 = 10$

$t_{s_8} = 53 \bmod 11 = 9$

$t_{s_9} = 35 \bmod 11 = 2$

```
if ts == p
    if P[1....m] == T[s+1..... s+m]
        print (pattern occurs with shift 's')
```

We can compute t_s using the following formula:

$$t_{s+1} = 10\left(t_s - 10^{m-1} T[s+1]\right) + T[s+m+1]$$

For $m = 2$ & $s = 0$, $t_s = 31$

We wish to remove higher order digit $T[s+1] = 3$ & bring the new lower order digit $T[s+m+1] = 4$

$t_{s+1} = 10(31 - 10 \cdot 3) + 4 = 14$

$t_{s+2} = 10(14 - 10 \cdot 1) + 1 = 41$