

Name: Atharva Bilonikar  
Uid No: 2021700011  
CSE DS D1  
DAA exp 10

Aim: String Matching algorithms (To implement Robin Karp algorithm)

Algorithm :

---

**Algorithm 1:** Naive String Searching Algorithm

---

**Data:** P: The pattern to look for  
T: The text to look in  
**Result:** Returns the number of occurrences of P in T

```
answer  $\leftarrow$  0;  
n  $\leftarrow$  length(T);  
k  $\leftarrow$  length(P);  
for i  $\leftarrow$  0 to n - k do  
    valid  $\leftarrow$  true;  
    for j  $\leftarrow$  0 to k - 1 do  
        if T[i + j]  $\neq$  P[j] then  
            valid  $\leftarrow$  false;  
            break;  
        end  
    end  
    if valid = true then  
        answer  $\leftarrow$  answer + 1;  
    end  
end  
return answer;
```

---

Code:

// Rabin-Karp algorithm in C

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

```
#define d 10
```

```
void rabinKarp(char pattern[], char text[], int q) {  
    int m = strlen(pattern);  
    int n = strlen(text);  
    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 + pattern[i]) % q;
t = (d * t + text[i]) % q;
}

for (i = 0; i <= n - m; i++) {
    if (p == t) {
        for (j = 0; j < m; j++) {
            if (text[i + j] != pattern[j])
                break;
        }

        if (j == m)
            printf("Pattern is found at position: %d \n", i + 1);
    }

    if (i < n - m) {
        t = (d * (t - text[i] * h) + text[i + m]) % q;

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

int main() {
    char text[50];
    char pattern[50];
    int q = 13;
    scanf("%s",text);
    scanf("%s",pattern);

    rabinKarp(pattern, text, q);
}

```

Output :

```

input
atharva
arva
Pattern is found at position: 4

...Program finished with exit code 0
Press ENTER to exit console.

```

Conclusion : In this experiment I understood Rabin karp algorithm