Lab 4

- **Aim : -** Implement program for string matching using Rabin-Karp algorithm.

- **Program : -**

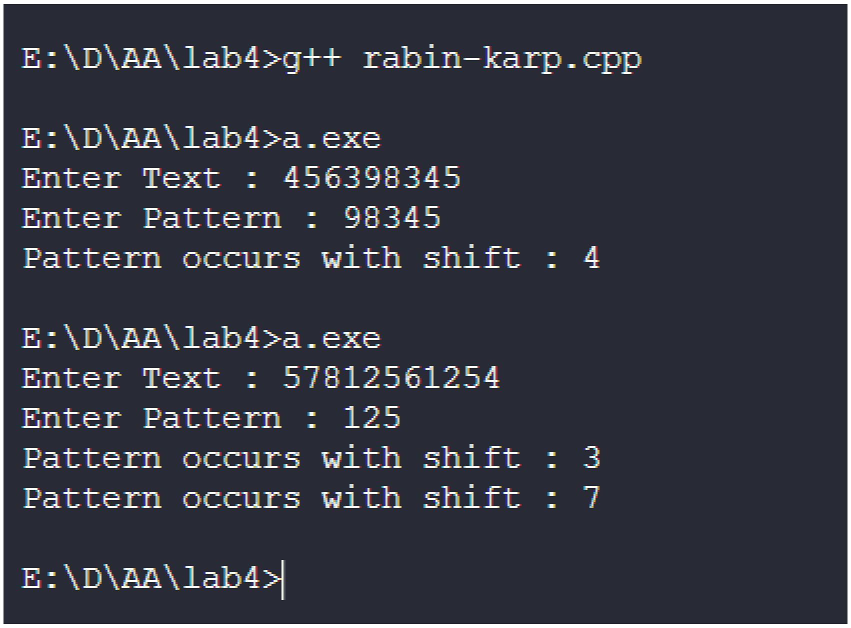
```
#include <bits/stdc++.h>
#define ll long long
using namespace std;

ll power(ll a,ll n,ll mod){
    ll res=1;
    a=a%mod;
    while(n>0){
        if(n&1)
            res=(res*a)%mod;
        n=n>>1;
        a=(a*a)%mod;
    }
    return res;
}

void rabinKarp(string T,string P,ll d,ll q){
    ll n=T.length(),m=P.length();
    ll t[n-m]={0};
    ll h=power(d,m-1,q),p=0,i;
    for(i=0;i<m;i++){
        p=(d*p+(P[i]-'0'))%q;
        t[0]=(d*t[0]+(T[i]-'0'))%q;
    }
    for(i=0;i<=(n-m);i++){
        if(p==t[i]){
            if(P==T.substr(i,m))
                cout<<"Pattern occurs with shift : "<<i<<"\n";
        }
        if(i<(n-m)){
            t[i+1]=((d*(t[i]-((T[i]-'0')*h))) + (T[i+m]-'0'))%q;
            if(t[i+1]<0)
                t[i+1]=t[i+1]+q;
        }
    }
}

int main(){
    string t,p;
    cout<<"Enter Text : ";
    cin>>t;
    cout<<"Enter Pattern : ";
    cin>>p;
    rabinKarp(t,p,10,13);
    return 0;
}
```

- **Output : -**



```
E:\D\AA\lab4>g++ rabin-karp.cpp

E:\D\AA\lab4>a.exe
Enter Text : 456398345
Enter Pattern : 98345
Pattern occurs with shift : 4

E:\D\AA\lab4>a.exe
Enter Text : 57812561254
Enter Pattern : 125
Pattern occurs with shift : 3
Pattern occurs with shift : 7

E:\D\AA\lab4>
```

- **Complexity of algorithm : -**

  - Running time of Rabin-Karp algorithm is approximately O(n+m).

  - In worst case, when there is valid shift possible at every position, time complexity of Rabin-Karp is same as Naive string matching algorithm and it is O((n-m+1)m).

  - Overall, this algo gives better performance as we are using hashing here and comparing the hash value of pattern & compare it with hash values of possible substrings of text string.