```cpp
//User function Template for C++

// d is the number of characters in the input alphabet
#define d 256

//Function to check if the pattern is present in string or not.
bool search(string pat, string txt, int q)
{
    int m = pat.size();
    int n = txt.size();
        //formuala is t1 = ((t0 - txt[i] * d^m-1)*d) + txt[i+m];
        // we will always %q every calculation to prevent stack overflow
        //calculating d^(m-1)
        int h = 1;
        for(int i = 0; i < m-1; i++) {
            h = (h*d) % q;
        }

        // calculating p(hash value of pattern) and t(hash value of text)

        int p = 0, t = 0;
        for(int i = 0; i < m; i++) {
            p = (p*d + pat[i]) % q;
            t = (t*d + txt[i]) % q;
        }

        //checking spurious hits

        for(int i = 0; i <= n-m; i++) {
            if(p == t) {
                int j;
                for(j = 0; j < m; j++) {
                    if(pat[j] != txt[j + i]) break;
                }
                if(j == m) return true;
            }
            //calculating t1 using t0; or rolling hash value
            if(i < n - m) {
                t = (((t - txt[i] * h) * d) + txt[i + m]) % q;
                //since the value of t can be negative to overcome this
                if(t < 0) t = t + q;
            }
        }
        return false;
}
```