Robin-Karp Hashing Algorithm

```
char text[NN],pattern[NN];
LL b1=311,b2=313,m1=1000000007,m2=1000000009;
LL call()  //count the number of times pattern occurs as a substring of text
{
    LL n=strlen(text),m=strlen(pattern),cnt=0;
    if(n < m) return 0;
    LL hp1=0,hp2=0;
    for(int i=0; i<m; i++)
    {
        hp1=(((hp1*b1)%m1)+pattern[i])%m1; //rolling hash
        hp2=(((hp2*b2)%m2)+pattern[i])%m2; //rolling hash
    }
    LL ht1=0,ht2=0,p1=1,p2=1;
    for(int i=0; i<m; i++)
    {
        ht1=(((ht1*b1)%m1)+text[i])%m1; //rolling hash
        ht2=(((ht2*b2)%m2)+text[i])%m2; //rolling hash
    }
    if(ht1==hp1 && ht2==hp2) cnt++;
    for(int i=0; i<m-1; i++)
    {
        p1=(p1*b1)%m1;
        p2=(p2*b2)%m2;
    }
    for(int i = m; i < n; i++)
    {
        ht1=(((ht1+m1-(text[i-m]*p1)%m1)*b1)%m1+text[i])%m1;
        ht2=(((ht2+m2-(text[i-m]*p2)%m2)*b2)%m2+text[i])%m2;
        if(ht1==hp1 && ht2==hp2) cnt++;
    }
    return cnt;
}
```