

```

81 Double Hash for number:
82 struct DoubleHash{
83     const int p1=53, p2=137; // 31 , 53 , 137 , 1331 , 1333
84     const int m1=1e9+9, m2=1e9+7; // 1e9+7 , 1e9+9
85     vector<ll> pow1, pow2, h1, h2;
86     int end = 0;
87
88     void build(int num) {
89         int n = 1;
90         end++;
91
92         pow1.resize(n);
93         h1.resize(n);
94         pow1[0]=1;
95         h1[0]=num+1;
96
97         pow2.resize(n);
98         h2.resize(n);
99         pow2[0]=1;
100        h2[0]=num+1;
101    }
102
103    void add(int num){
104        end++;
105        pow1.push_back(pow1.back()*p1%m1);
106        h1.push_back((h1.back()*p1 + num+1)%m1);
107
108        pow2.push_back(pow2.back()*p2%m2);
109        h2.push_back((h2.back()*p2 + num+1)%m2);
110    }
111
112    pair<int, int> getHash(int i, int j){
113        ll a = h1[j];
114        if(i) a -= h1[i-1] * pow1[j-i+1];
115        a = (a%m1 + m1)%m1;
116
117        ll b = h2[j];
118        if(i) b -= h2[i-1] * pow2[j-i+1];
119        b = (b%m2 + m2)%m2;
120
121        return make_pair(a,b);
122    }
123 };

```