**find\_by\_order(k)** – ফাংশনটি **kth ordered element** এর একটা পয়েন্টার রিটার্ন করে। অর্থাৎ তুমি চাইলেই kth ইন্ডেক্সে কি আছে, সেটা জেনে ফেলতে পারছো!

**order\_of\_key(x)** – ফাংশনটি **x** এলিমেন্টটা কোন পজিশনে আছে সেটা বলে দেয়।

#include <bits/stdc++.h>

#include <ext/pb\_ds/tree\_policy.hpp>

#include <ext/pb\_ds/assoc\_container.hpp>

using namespace std;

using namespace \_\_gnu\_pbds;

template <typename T> using orderedSet =

    tree<T, null\_type, less<T>,

    rb\_tree\_tag, tree\_order\_statistics\_node\_update>;

int main()

{

    int n,k;

    cin >> n >> k;

    orderedSet<int> s;

    orderedSet<int>::iterator it;

    for(int i = 1; i <= n; i++)

        s.insert(i);

    int cur = 0;

    while(n)

    {

        cur = (cur + k - 1) % n;

        it = s.find\_by\_order(cur);

        cout << \*it << endl;

        s.erase(it);

        n--;

    }

    return 0;

}

|  |
| --- |
| orderedSet<int> os;    os.insert(5);  os.insert(2);  os.insert(6);  os.insert(4);  os.insert(2);  os.insert(7);    cout << \*os.find\_by\_order(0)<<endl; // 2  cout << \*os.find\_by\_order(1)<<endl; // 4  cout << \*os.find\_by\_order(2)<<endl; // 5  cout << \*os.find\_by\_order(3)<<endl; // 6  cout << \*os.find\_by\_order(4)<<endl; // 7    cout << os.order\_of\_key(5) << endl;  // 2  cout << os.order\_of\_key(2) << endl;  // 0  cout << os.order\_of\_key(6) << endl;  // 3  cout << os.order\_of\_key(4) << endl;  // 1  cout << os.order\_of\_key(7) << endl;  // 4 |