

# Top k most frequent elements in the stream

## Problem

Given an array  $a[]$  of size  $n$  and value  $k$ . Our task is to output elements in decreasing frequency till we reach  $(k+1)^{\text{th}}$  distinct element.

Example:

1	2	2	2	3	2
---	---	---	---	---	---

and  $k = 2$

Output:

Key		Value
1	→	1
2	→	3

## Approach

1. Create a map (say *freq*)

`map<int,int> freq;`

2. While traversing the array keep track of elements and when we find  $(k+1)^{\text{th}}$  distinct element, break from the loop.
3. Output the element and frequency using map traversal.

## Code

```
int main(){
    int n, k;
    cin >> n >> k;

    vector<int> a(n);

    for(int i=0; i<n; i++) {
        cin >> a[i];
    }
    map<int,int> freq;
    int cnt=0;
    for(int i=0; i<n; i++) {
        if(cnt == k && freq[a[i]] == 0){
            break;
        }
        if(freq[a[i]] == 0){
            cnt++;
        }
        freq[a[i]]++;
    }

    map<int,int> :: iterator it;
    for(it = freq.begin(); it!=freq.end(); it++)
    {
        if(it->second != 0){
            cout << it->first << " " << it->second << endl;
        }
    }
    return 0;
}
```