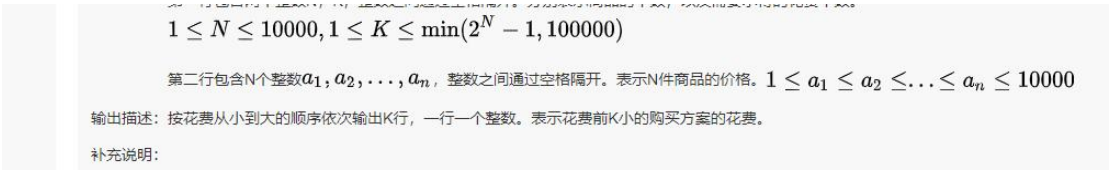
C++-数组贪心堆-商店里有 N 件唯一性商品
题目描述：
商店里有 N 件唯一性商品，每件商品有一个价格，第 i 件商品的价格是 a_i。一个购买方案可以是从 N 件商品中选择任意件进行购买（至少一件），花费即价格之和。现在你需要求出所有购买方案中花费前 K 小的方案，输出这些方案的花费。

当两个方案选择的商品集合内至少有一件不同，视为不同方案，因此可能存在两个方案花费相同。

输入描述：
输入数据含三行

第一行包含两个整数 N，K，整数之间通过空格隔开。分别表示商品的个数，以及需要求得的花费个数。


$1 \leq N \leq 10000, 1 \leq K \leq \min(2^N - 1, 100000)$

第二行包含N个整数$a_1, a_2, \ldots, a_n$，整数之间通过空格隔开。表示N件商品的价格。$1 \leq a_1 \leq a_2 \leq \ldots \leq a_n \leq 10000$

输出描述：按花费从小到大的顺序依次输出K行，一行一个整数。表示花费前K小的购买方案的花费。

补充说明：

示例 1
输入：
5 6
1 1 2 3 3
输出：
1
1
2
2
3
3
说明：

花费前 10 小的方案：

集合	花费
1	1
2	1
3	2
1,2	2
4	3
5	3
1,3	3
2,3	3
1,2,3	4
1,4	4

集合　　花费
1　1
2　1
3　2
1,2　2
4　3
5　3
1,3　3
2,3　3
1,2,3　　4
1,4　4
示例 2
输入：
4 15
1 2 3 4
输出：
1
2
3
3
4
4
5
5

6
6
7
7
8
9
10
说明：
示例 3
输入：
3 7
1 10 100
输出：
1
10
11
100
101
110
111
说明：

```
#include <iostream>
#include <vector>
#include <queue>

using namespace std;

struct node{
	long long val;
	long long loc;
	node(long long v, long long l){
		val = v;
		loc = l;
	}
};

struct cmp{
	bool operator()(node* a, node* b){
		return a->val > b->val;
	}
};

int main() {
	ios::sync_with_stdio(false);
```

```
    int N,K;
    cin>> N >> K;
    vector<long long> vec(N);
    for(int i=0; i<N; i++){
        cin >> vec[i];
    }
    priority_queue <node*, vector<node*>,cmp> pq;
    pq.push(new node(vec[0],0));
    while(K--){
        node* tmp = pq.top();
        cout<< tmp->val<< endl;
        pq.pop();
        if(tmp->loc +1 < vec.size()){
            pq.push(new node(tmp->val + vec[tmp->loc+1], tmp->loc+1));
            pq.push(new node(tmp->val + vec[tmp->loc+1] - vec[tmp->loc],tmp->loc+1));
        }
    }
    return 0;
}
// 64 位输出请用 printf("%lld")
```