**#include <bits/stdc++.h>**

**using namespace std;**

**#define M 1000000007**

**void solve() {**

**int n;**

**cin>>n;**

**priority\_queue<int, vector<int>, greater<int>> pq;**

**for(int i=0;i<n;i++) {**

**int x;**

**cin>>x;**

**pq.push(x);**

**}**

**long long ans = 0;**

**while(pq.size() > 1) {**

**int e1 = pq.top();**

**pq.pop();**

**int e2 = pq.top();**

**pq.pop();**

**ans += (0ll + e1 + e2 - 1);**

**pq.push(e1 + e2);**

**}**

**cout<<ans<<endl;**

**}**

**int main() {**

**/\* int t;**

**cin>>t;**

**while(t--) {**

**solve();**

**}\*/**

**// Merging k sorted Array**

**int numberArrays;**

**cin>>numberArrays;**

**vector<vector<int>>arr2d(numberArrays);**

**priority\_queue<pair<int, pair<int, int>>>pq;**

**int totalSize = 0;**

**for(int i=0;i<numberArrays;i++) {**

**int n;**

**cin>>n;**

**arr2d[i].resize(n);**

**for(int j=0;j<n;j++) {**

**cin>>arr2d[i][j];**

**if(j==0) {**

**pq.push({-1\*arr2d[i][j], {0, i}});**

**}**

**}**

**totalSize+=n;**

**}**

**vector<int>ans(totalSize);**

**for(int i=0;i<totalSize;i++) {**

**auto p = pq.top();**

**pq.pop();**

**ans[i] = -1\*p.first;**

**if(p.second.first + 1 < arr2d[p.second.second].size() ) {**

**pq.push({-1\*arr2d[p.second.second][p.second.first + 1], {p.second.first + 1, p.second.second}});**

**}**

**}**

**for(int i=0;i<totalSize;i++) {**

**cout<<ans[i]<<" ";**

**}**

**cout<<endl;**

**return 0;**

**}**