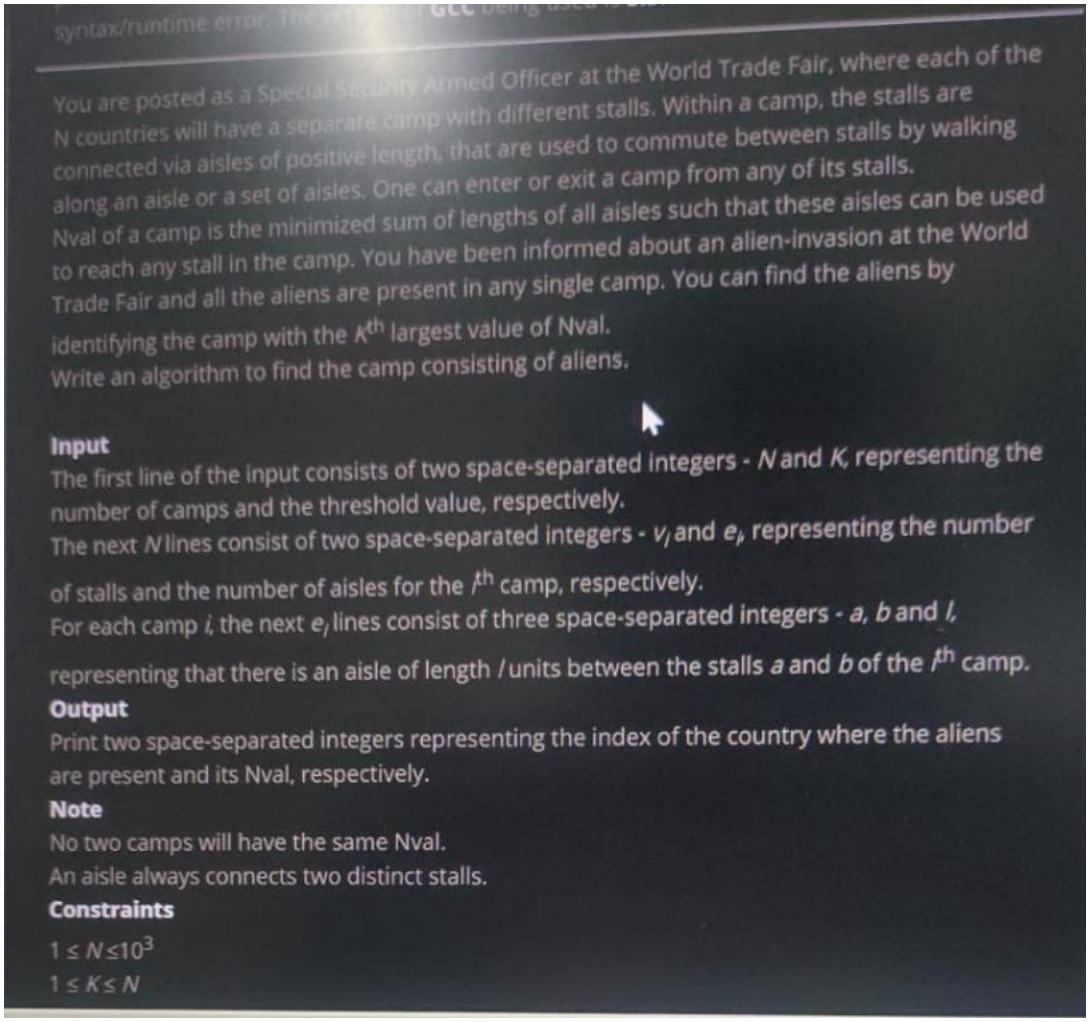
FLIPKART

1)



syntax/runtime error. The [illegible] GCC being [illegible]

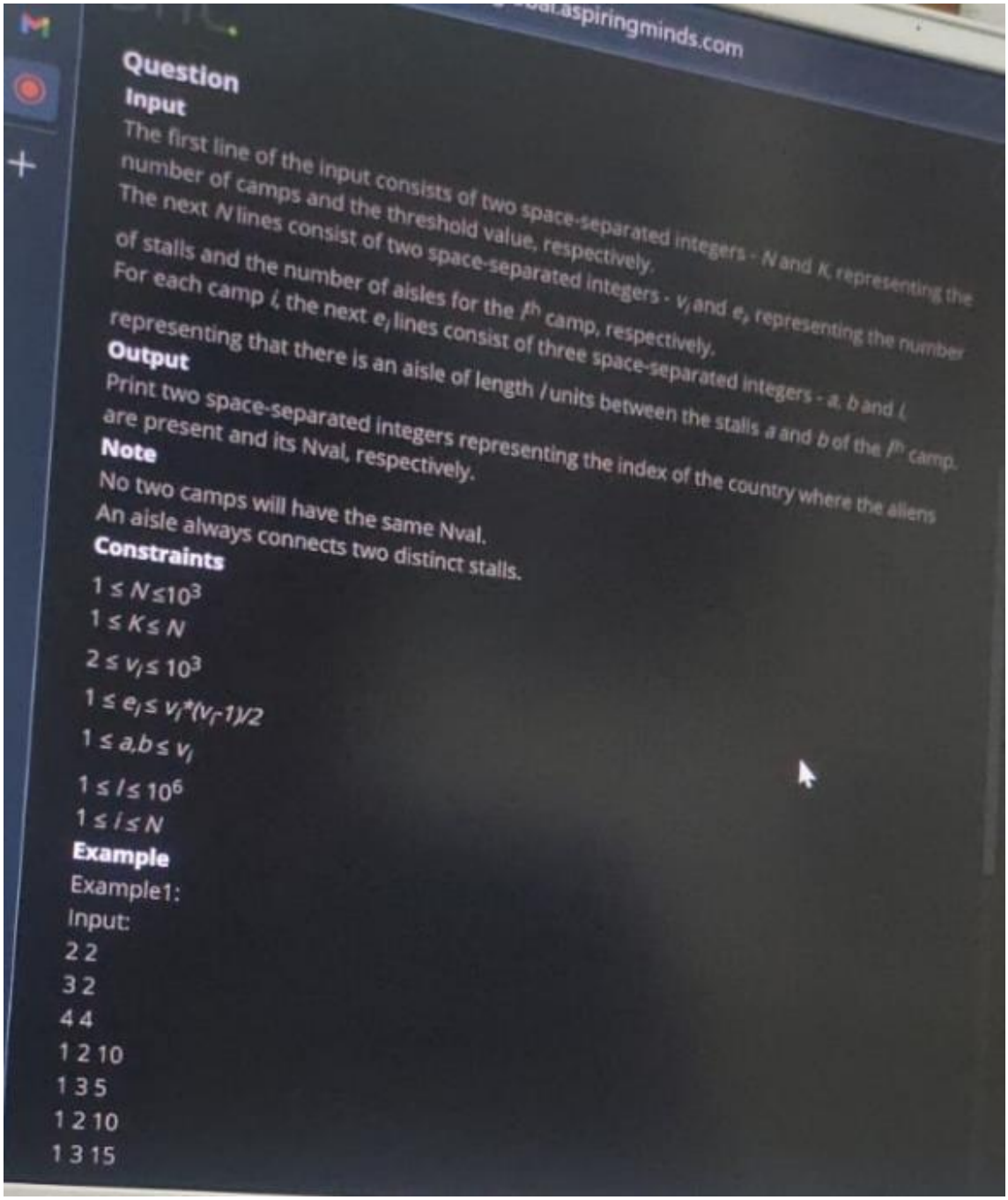
You are posted as a Special Security Armed Officer at the World Trade Fair, where each of the N countries will have a separate camp with different stalls. Within a camp, the stalls are connected via aisles of positive length, that are used to commute between stalls by walking along an aisle or a set of aisles. One can enter or exit a camp from any of its stalls. Nval of a camp is the minimized sum of lengths of all aisles such that these aisles can be used to reach any stall in the camp. You have been informed about an alien-invasion at the World Trade Fair and all the aliens are present in any single camp. You can find the aliens by identifying the camp with the K^{th} largest value of Nval.
Write an algorithm to find the camp consisting of aliens.

Input
The first line of the input consists of two space-separated integers - N and K, representing the number of camps and the threshold value, respectively.
The next N lines consist of two space-separated integers - v_i and e_i, representing the number of stalls and the number of aisles for the i^{th} camp, respectively.
For each camp i, the next e_i lines consist of three space-separated integers - a, b and l, representing that there is an aisle of length l units between the stalls a and b of the i^{th} camp.

Output
Print two space-separated integers representing the index of the country where the aliens are present and its Nval, respectively.

Note
No two camps will have the same Nval.
An aisle always connects two distinct stalls.

Constraints
$1 \le N \le 10^3$
$1 \le K \le N$


aspiringminds.com
Question
Input
The first line of the input consists of two space-separated integers - N and K, representing the number of camps and the threshold value, respectively.
The next N lines consist of two space-separated integers - v_i and e_i, representing the number of stalls and the number of aisles for the i^{th} camp, respectively.
For each camp i, the next e_i lines consist of three space-separated integers - a, b and l, representing that there is an aisle of length l units between the stalls a and b of the i^{th} camp.
Output
Print two space-separated integers representing the index of the country where the aliens are present and its Nval, respectively.
Note
No two camps will have the same Nval.
An aisle always connects two distinct stalls.
Constraints
$1 \le N \le 10^3$
$1 \le K \le N$
$2 \le v_i \le 10^3$
$1 \le e_i \le v_i*(v_i-1)/2$
$1 \le a,b \le v_i$
$1 \le l \le 10^6$
$1 \le i \le N$
Example
Example1:
Input:
2 2
3 2
4 4
1 2 10
1 3 5
1 2 10
1 3 15

Question

$1 \leq i \leq N$

Example

Example1:

Input:

2 2

3 2

4 4

1 2 10

1 3 5

1 2 10

1 3 15

2 4 5

3 4 20

Output:

1 15

Explanation:

The Nval for camp-1 is 15 and the Nval for camp-2 is 30.

The second largest Nval is for camp-1 which is 15.

So, the output is (1 15).

Example2:

Input:

2 1

4 4

4 3

1 2 10

1 3 15

2 4 5

3 4 20

1 2 10

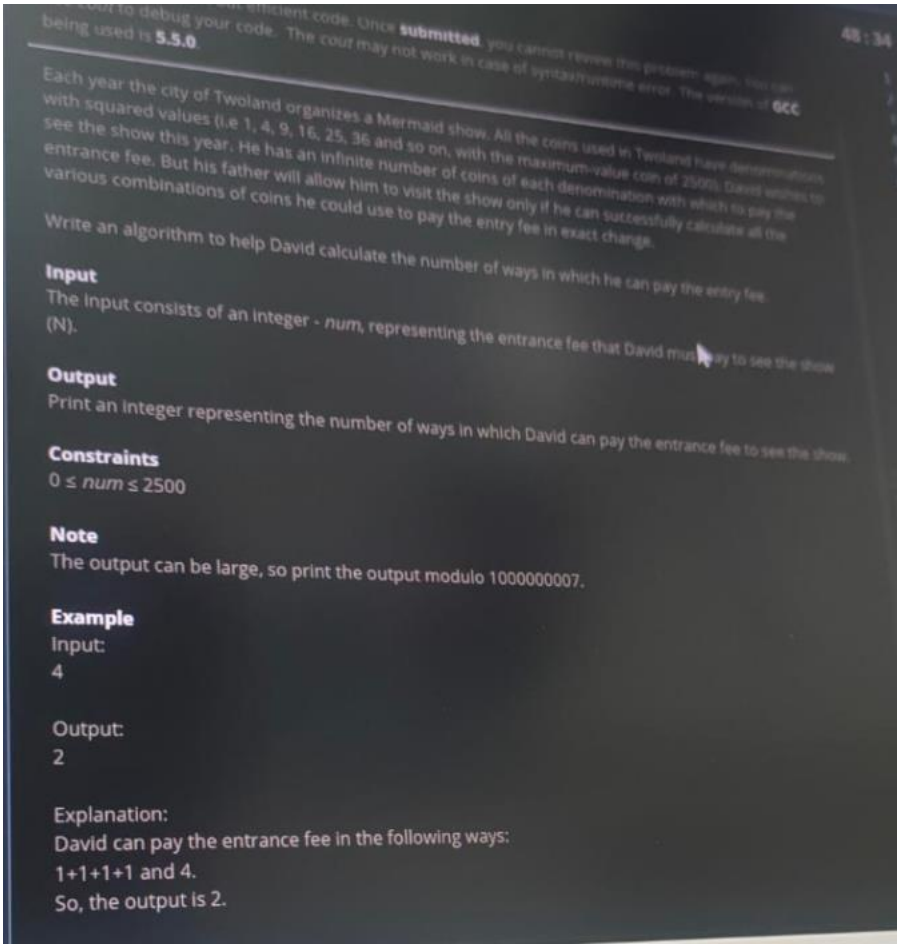
1 3 5

1 4 20

Output:

2 35

2)



efficient code. Once **submitted**, you cannot review this problem again. You can use cout to debug your code. The cout may not work in case of syntax/runtime error. The version of **GCC** being used is **5.5.0**

48 : 34

Each year the city of Twoland organizes a Mermaid show. All the coins used in Twoland have denominations with squared values (i.e 1, 4, 9, 16, 25, 36 and so on, with the maximum-value coin of 2500). David wishes to see the show this year. He has an infinite number of coins of each denomination with which to pay the entrance fee. But his father will allow him to visit the show only if he can successfully calculate all the various combinations of coins he could use to pay the entry fee in exact change.

Write an algorithm to help David calculate the number of ways in which he can pay the entry fee.

Input
The input consists of an integer - *num*, representing the entrance fee that David must pay to see the show (N).

Output
Print an integer representing the number of ways in which David can pay the entrance fee to see the show.

Constraints
$0 \leq num \leq 2500$

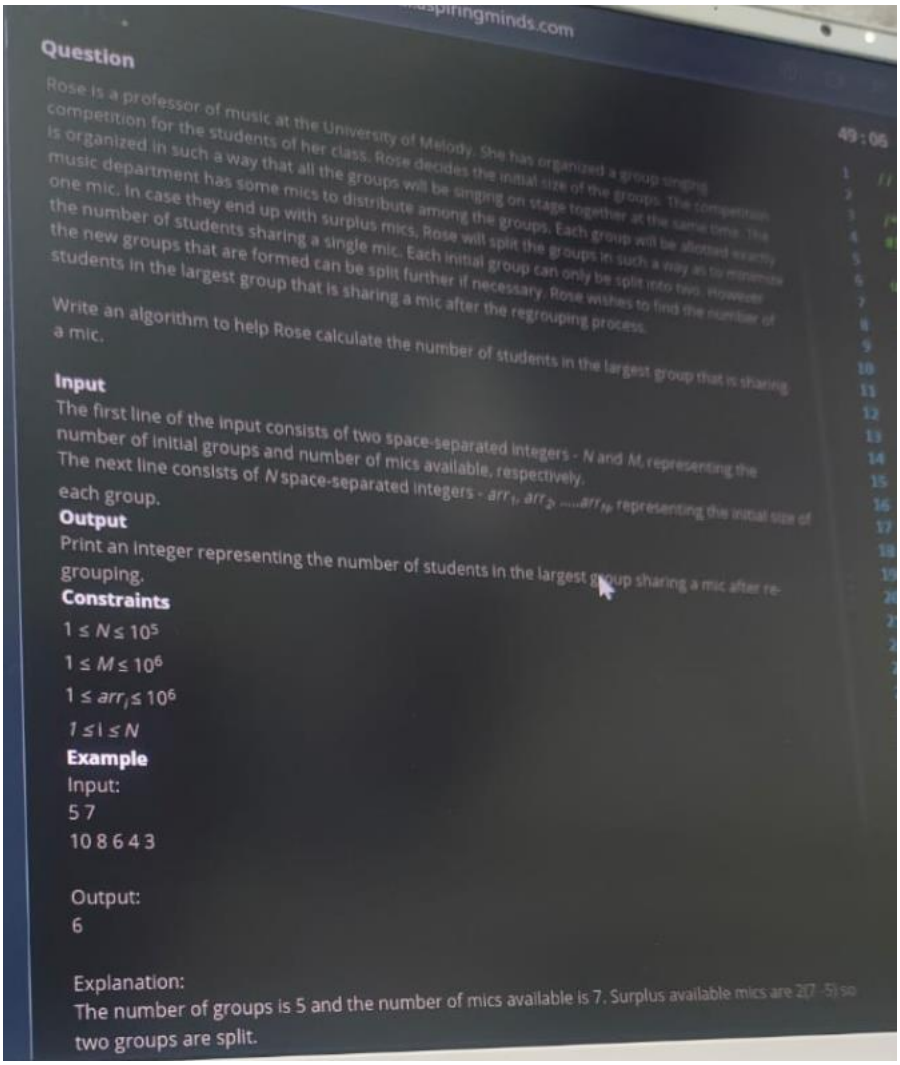
Note
The output can be large, so print the output modulo 1000000007.

Example
Input:
4

Output:
2

Explanation:
David can pay the entrance fee in the following ways:
1+1+1+1 and 4.
So, the output is 2.

3)



aspiringminds.com

Question

Rose is a professor of music at the University of Melody. She has organized a group singing competition for the students of her class. Rose decides the initial size of the groups. The competition is organized in such a way that all the groups will be singing on stage together at the same time. The music department has some mics to distribute among the groups. Each group will be allotted exactly one mic. In case they end up with surplus mics, Rose will split the groups in such a way as to minimize the number of students sharing a single mic. Each initial group can only be split into two. However the new groups that are formed can be split further if necessary. Rose wishes to find the number of students in the largest group that is sharing a mic after the regrouping process.

Write an algorithm to help Rose calculate the number of students in the largest group that is sharing a mic.

Input

The first line of the input consists of two space-separated integers - N and M, representing the number of initial groups and number of mics available, respectively.
The next line consists of N space-separated integers - $arr_1, arr_2, \ldots, arr_N$ representing the initial size of each group.

Output

Print an integer representing the number of students in the largest group sharing a mic after re-grouping.

Constraints

$1 \le N \le 10^5$

$1 \le M \le 10^6$

$1 \le arr_i \le 10^6$

$1 \le i \le N$

Example

Input:
5 7
10 8 6 4 3

Output:
6

Explanation:
The number of groups is 5 and the number of mics available is 7. Surplus available mics are 2(7 -5) so two groups are split.

Solution

1)

```
#include<iostream>
#include<bits/stdc++.h>
#define ll long long
using namespace std;

int findMin(vector<int> &key,vector<bool> &mst,int n)
{
    int mini=INT_MAX,ind=0;
    for(int i=0;i<n;i++)
    {
        if(key[i]<mini && !mst[i])
        mini=key[i],ind=i;
    }
    return ind;
}

ll prims(vector<vector<pair<ll,ll>>> &adj,int n,int m)
{
    vector<int> key(n),parent(n);
    vector<bool> mst(n,false);

    for(int i=0;i<n;i++)
    key[i]=INT_MAX, parent[i]=-1;

    key[0]=0;
    int cost=0;
    for(int i=0;i<m+1;i++)
    {
        int u=findMin(key,mst,n);
        mst[u]=true;
        cost+=key[u];
        for(auto &it:adj[u])
        {
            if(!mst[it.first] && it.second<key[it.first])
            parent[it.first]=u,key[it.first]=it.second;
        }
    }
    return cost;
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    ll n,k;
    cin>>n>>k;
    vector<pair<ll,ll>> camp(n);
    for(ll i=0;i<n;i++)
    {
        ll v,e;
        cin>>v>>e;
        camp[i].first=v;
        camp[i].second=e;
```

```
    }
    vector<pair<ll,ll>> nval(n);
    for(ll i=0;i<n;i++)
    {
        vector<vector<pair<ll,ll>>> adj(camp[i].first);
        for(ll j=0;j<camp[i].second;j++)
        {
            ll a,b,l;
            cin>>a>>b>>l;
            adj[a-1].push_back({b-1,l});
            adj[b-1].push_back({a-1,l});
        }
        nval[i]={prims(adj,camp[i].first,camp[i].second),i+1};
    }
    sort(nval.begin(),nval.end());
    cout<<nval[n-1-k+1].second<<" "<<nval[n-1-k+1].first;
    return 0;

}
```

2)

```
int mod = 1e9 + 7;

int func(vector<int> &sq, int i, int sum, vector<vector<long long int>> &dp)
{
    if (sum == 0)
        return 1;

    if (i == sq.size())
        return 0;

    if (dp[sum][i] != -1)
        return dp[sum][i];

    long long int ans;
    if (sq[i] <= sum)
    {
        ans = (func(sq, i, sum - sq[i], dp) % mod + func(sq, i + 1, sum, dp) % mod) %
mod;
    }
    else
    {
        ans = 0;
    }

    return dp[sum][i] = ans % mod;
}

void solve()
{
    int n;
    cin >> n;

    vector<int> sq;
```

```
    for (int i = 1; i <= 50; i++)
    {
        sq.pb(i * i);
    }

    vector<vector<long long int>> dp(n + 10, vector<long long int>((int)sq.size() +
10, -1));

    cout << func(sq, 0, n, dp) << "\n";
}
```

3)

```
#include<bits/stdc++.h>
#define ll long long
#define S second
#define F first
using namespace std;

int check(vector<int> &v, int n, int mid, int k)
{
    int cnt =0;
    for(int i=0; i<n; ++i)
        cnt += (v[i]-1)/mid;
    return cnt <= k;
}

int main()
{
    int t = 1;
    // cin>>t;
    while(t--)
    {
        int n, m;
        cin>>n>>m;
        vector<int> arr;
        int mx = -1e9;
        for(int i=0; i<n; ++i)
            cin>>arr[i], mx = max(mx, arr[i]);
        int k = m - n;
        int l = 1, r = mx;
        while(l<r){
            int mid = (l+r)/2;
            if(check(arr, n, mid, k)) r= mid;
            else l = mid+1;
        }
        cout<<r<<endl;
    }

    return 0;
}
```