// 并查集

#include <cstdio>

int const MAX = 30000 + 5;

int fa[MAX];

void UF\_set(int n)

{

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

fa[i] = i;}

int Find(int x)

{return x == fa[x] ? x : fa[x] = Find(fa[x]);}

void Union(int a, int b)

{

int r1 = Find(a);

int r2 = Find(b);

if(r1 == 0) fa[r2] = r1;

else if(r2 == 0) fa[r1] = r2;

else fa[r1] = r2;

}

int main()

{

int n, m, ans;

while(scanf("%d %d", &n, &m) != EOF && (n + m))

{

ans = 0;

UF\_set(n);

for(int i = 0; i < m; i++)

{

int get, fir, k;

scanf("%d %d", &k, &fir);

for(int j = 1; j < k; j++)

{

scanf("%d", &get);

Union(get, fir);

}

}

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

if(Find(i) == 0)

ans++;

printf("%d\n", ans);

int pre[maxn],sum[maxn];//sum[i]表示到根节点的总和

int n,m;

int find(int x)

{

if(x!=pre[x])

{

int f=pre[x];

pre[x]=find(pre[x]);

sum[x]+=sum[f];//求x到x集合中更节电的距离.

}

return pre[x];

}

int main()

{

int l,r,s;

while(~scanf("%d%d",&n,&m))

{

REPF(i,0,n) pre[i]=i;

CLEAR(sum,0);

int ans=0;

while(m--)

{

scanf("%d%d%d",&l,&r,&s);

l--;

int ra=find(l),rb=find(r);

cout << ra << ' ' << rb << endl;

if(ra!=rb)

{

pre[ra]=rb;//以rb作为根节点

sum[ra]=sum[r]-sum[l]+s;//sum[r]:r->rb;s:l->r;sum[l]:l->ra，要求ra->rb就是啦

}

else if(ra==rb)

{

if(sum[l]-sum[r]!=s) ans++;

}

}

printf("%d\n",ans); }