//**缩点**

#include <bits/stdc++.h>

**using** **namespace** std;

**const** **int** N = 4e6 + 50;

**int** head[N], etop;

**struct** Edge

{

**int** v, nxt;

}e[N \* 2];

**int** fr[N], to[N], m;

**void** add(**int** u, **int** v, **int** op = 0)

{

**if** (op)

{

++m;

fr[m] = u; to[m] = v;

}

e[++etop] = Edge{v, head[u]};

head[u] = etop;

}

**int** low[N], num, st[N], top, dfn[N], val[N], bl[N], cnt;

**void** tarjan(**int** u)

{

dfn[u] = low[u] = ++num;

st[++top] = u;

**for** (**int** v, i = head[u]; i; i = e[i].nxt)

{

v = e[i].v;

**if** (!dfn[v])

{

tarjan(v);

low[u] = min(low[u], low[v]);

}

**else** **if** (!bl[v]) low[u] = min(low[u], dfn[v]);

}

**if** (low[u] == dfn[u])

{

++cnt;

**while** (st[top + 1] != u)

{

**int** x = st[top--];

bl[x] = cnt;

++val[cnt];

}

}

}

**int** f[N];

**void** dfs(**int** u)

{

f[u] = val[u];

**for** (**int** v, i = head[u]; i; i = e[i].nxt)

{

v = e[i].v;

**if** (!f[v]) dfs(v);

f[u] += f[v];

}

}

**int** n, M;

map <**int**, **int**> mp[N];

**int** main()

{

scanf("%d%d", &n, &M);

**for** (**int** i = 1; i <= M; i++)

{

**int** pre = 0;

**for** (**int** x, j = 1; j <= n; j++)

{

scanf("%d", &x);

**if** (j >= 2) add(pre, x, 1);

pre = x;

}

}

**for** (**int** i = 1; i <= n; i++)

**if** (!dfn[i]) tarjan(i);

memset(head, 0, **sizeof**(head));

etop = 0;

**for** (**int** i = 1; i <= m; i++)

**if** (bl[fr[i]] != bl[to[i]])

{

**int** u = bl[fr[i]], v = bl[to[i]];

**if** (mp[u][v]) **continue**;

mp[u][v] = 1;

*//cout<<u<<" "<<v<<endl;*

add(u, v);

}

**for** (**int** i = 1; i <= cnt; i++)

**if** (!f[i]) dfs(i);

**for** (**int** i = 1; i <= n; i++)

printf("%d ", f[bl[i]] - 1);

**return** 0;

*//以一个点为结尾的回文串个数*

#include <bits/stdc++.h>

**using** **namespace** std;

**const** **int** N = 1e6 + 50;

**int** ans;

**struct** PAM

{

**int** f[N], len[N], top, n, last;

**int** ch[N][26], s[N], cnt[N];

PAM ()

{

s[0] = -1; f[0] = 1;

len[1] = -1; top = 1;

}

**int** getfail(**int** u)

{

**while** (s[n - len[u] - 1] != s[n])

u = f[u];

**return** u;

}

**void** add(**int** c)

{

s[++n] = c;

**int** u = getfail(last);

**int** &v = ch[u][c];

**if** (!v)

{

**int** np = ++top;

len[np] = len[u] + 2;

f[np] = ch[getfail(f[u])][c];

cnt[np] = cnt[f[np]] + 1;

v = np;

}

last = v;

ans = cnt[last];

printf("%d ", ans);

}

}pam;

**char** c;

**int** main()

{

**while** (c = getchar())

{

**if** (c < 'a' || c > 'z') **break**;

c = (c - 97 + ans) % 26 + 97;

pam.add(c);

}

**return** 0;

}