**一、原程序（用cin,cout代替scanf、printf）**

#include <iostream>

#include <cstdio> //scanf,printf

using namespace std;

const int LP=100001;

int n,m,k;

int x[LP],y[LP],d[LP],c[LP],\*a[LP];

int main()

{

cin>>n>>m>>k;

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

{

cin>>x[i]>>y[i]>>d[i];

c[y[i]]++;

}

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

a[i] = new int[c[i]];

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

{

\*a[y[i]] = d[i];

a[y[i]]++;

}

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

{

a[i]-=c[i];

for(int j=1;j<=c[i];j++,a[i]++)

cout<<\*a[i]<<" ";

}

return 0;

}

**二、空间改进（用x代替x[]，用k+1,m+1代替100001）**

#include <iostream>

#include <cstdio> //scanf,printf

#include <cstring>

using namespace std;

int main()

{

int n,m,k;

cin>>n>>m>>k;

int LP=k+1;

int x,y[LP],d[LP],c[m+1]={0},\*a[m+1];

//memset(c,0,sizeof(c));

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

{

cin>>x>>y[i]>>d[i];

c[y[i]]++;

}

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

a[i] = new int[c[i]];

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

{

\*a[y[i]] = d[i];

a[y[i]]++;

}

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

{

a[i]-=c[i];

for(int j=1;j<=c[i];j++,a[i]++)

cout<<\*a[i]<<" ";

}

return 0;

}

**三、空间改进（用二维数组代替两个一维数组，充分利用了第0列）**

#include <iostream>

using namespace std;

int main()

{

int n,m,k,temp;

int x,y,data;

cin>>n>>m>>k;

int a[m+1][k+1]={0};

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

{

cin>>x>>y>>data;

temp = a[y][0]+1;

a[y][temp]=data;

a[y][0]++;

}

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

{

for(int j=1;j<=(a[i])[0];j++)

cout<<a[i][j]<<" ";

}

return 0;

}

**四、终极版（只用两个一维数组，遍历扫描输出，代码行最少，从21行减至13行）**

#include <iostream>

using namespace std;

int main()

{

int n,m,k;

//输入n,m,k

cin>>n>>m>>k;

//动态定义数组

int LP=k+1;

int x,y[LP],d[LP];

//记录后两数据，x值不记录

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

{

cin>>x>>y[i]>>d[i];

}

//遍历输出数据

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

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

if(y[j]==i) cout<<d[j]<<" ";

return 0;

}