#include <iostream>

#include<bits/stdc++.h>

//dijkstras algo for adj matrix

using namespace std;

bool visited[10000];

int dist[10000];

int arr[1000][1000];

int n,e,i,j;

int findminvertex()

{

int p=INT\_MAX,ans;

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

{

if(visited[i]==false&&dist[i]<p)

{

p=dist[i];

ans=i;

}

}

return(ans);

}

void dijkstra()

{

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

{

dist[i]=INT\_MAX;

visited[i]=false;

}

dist[0]=0;

for(int i=0;i<n-1;i++)

{

int minvertex=findminvertex();

visited[minvertex]=true;

for(j=0;j<n;j++)

{

if(visited[j]==false&&arr[minvertex][j]!=0)

{

int d=dist[minvertex]+arr[minvertex][j];//calc dist if reached through this minvertex

// now we need to check if this distance is lesser than the existing distance for j. if it is, we need to update.

dist[j]=min(d,dist[j]);

}

}

}

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

{

cout<<dist[i];

}

}

int main()

{

cin>>n>>e;

int x,y,wt;

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

{

cin>>x>>y>>wt;

arr[x][y]=wt;

}

dijkstra();

return 0;

}