```cpp
#include<bits/stdc++.h>
using namespace std;
typedef pair<int,int> P;
typedef long long ll;
const int inf = numeric_limits<int>::max()/2;

struct edge {int to; int cap; int rev;};
vector< vector< edge > > G;
int visit[1010];

void add_edge(int from,int to,int cap){
  edge e1,e2;
  e1.to = to;e2.to = from;
  e1.cap = cap;e2.cap = 0;
  e1.rev = (int)G[to].size();
  e2.rev = (int)G[from].size();
  G[from].push_back(e1);
  G[to].push_back(e2);
}

int dfs(int node,int flow,int target){
  visit[node] = 1;
  if(node == target)return flow;
  for(int i = 0;i < (int)G[node].size();++i){
    edge e = G[node][i];
    if(!visit[e.to] && e.cap > 0){
      int nflow = dfs(e.to,min(flow,e.cap),target);
      if(nflow){
        G[node][i].cap -= nflow;
        G[e.to][e.rev].cap += nflow;
        return nflow;
      }
    }
  }
  return 0;
}

int main(void){
  int v,e;
  cin >> v >> e;
  G.clear();
  G.resize(v);
  for(int i = 0;i < e;++i){
    int u,v,c;
    cin >> u >> v >> c;
    add_edge(u,v,c);
  }
  int res = 0;
  while(true){
    fill(visit,visit+1010,0);
    int d = dfs(0,inf,v-1);
    if(!d)break;
    res += d;
  }
  cout << res << endl;
  return 0;
}
```