Pastebin link: http://paste.ubuntu.com/24751541/

#define NODES 105  
#define INF 1001  
**using** **namespace** std;  
**int** res[NODES][NODES], cap[NODES][NODES], prev[NODES];  
**int** src, sink;  
**bool** vis[NODES];  
**int** n;  
**bool** bfs(){  
 memset(vis, **false**, **sizeof**(vis));  
 vis[src]=**true**;  
 queue <**int**> q;  
 q.push(src);  
 **while**(!q.empty()){  
 **int** u=q.front();  
 q.pop();  
 **for**(**int** i=1; i<=n; i++){  
 **if**(res[u][i]!=0 && !vis[i]){  
 vis[i]=**true**;  
 prev[i]=u;  
 **if**(i==sink) **return** **true**;  
 q.push(i);  
   
 }  
 }  
 }  
 **return** **false**;  
}  
  
**int** path\_cost(){  
 **int** cur=sink;  
 **int** ans=INF;  
 **while**(cur!=src){  
 ans=min(ans, res[prev[cur]][cur]);  
 cur=prev[cur];  
 }  
 **return** ans;  
}  
  
**void** augment\_path(**int** val){  
 **int** cur=sink;  
 **while**(cur!=src){  
 res[prev[cur]][cur]-=val;  
 res[cur][prev[cur]]+=val;  
 cur=prev[cur];  
 }  
}  
  
**int** max\_flow(){  
 **int** ans=0, val;  
 **while**(bfs()){  
 val=path\_cost();  
 augment\_path(val);  
 ans+=val;  
   
 }  
 **return** ans;  
}