```cpp
#include <bits/stdc++.h>
const int MAX_N = 1000;
const int MAX_V = 2*(MAX_N+1);
const int S = MAX_V - 2;
const int E = MAX_V - 1;
const int INF = 987654321;
using namespace std;

int N, M;
int C[MAX_V][MAX_V];
int D[MAX_V][MAX_V];
int F[MAX_V][MAX_V];
vector<int> graph[MAX_V];

int main(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    #endif // ONLINE_JUDGE

    scanf("%d %d", &N, &M);
    for (int i=0;i<N;i++){
        graph[i*2].push_back(i*2+1);
        graph[i*2+1].push_back(i*2);
        C[i*2][i*2+1] = INF;
    }
    graph[S].push_back(0);
    graph[0].push_back(S);
    C[S][0] = 2;
    graph[2*(N-1)+1].push_back(E);
    graph[E].push_back(2*(N-1)+1);
    C[2*(N-1)+1][E] = 2;

    for (int i=0;i<M;i++){
        int u, v, d;
        scanf("%d %d %d", &u, &v, &d);
        u--; v--;
        graph[u*2+1].push_back(v*2);
        graph[v*2].push_back(u*2+1);
        graph[v*2+1].push_back(u*2);
        graph[u*2].push_back(v*2+1);

        C[u*2+1][v*2] = 1;
        C[v*2+1][u*2] = 1;

        D[u*2+1][v*2] = d;
        D[v*2][u*2+1] = -d;
        D[v*2+1][u*2] = d;
        D[u*2][v*2+1] = -d;
    }

    int cost = 0;
    int xx = 2;
    while(xx--){
        int prev[MAX_V], dist[MAX_V];
        bool in_Q[MAX_V] = {0,};
        queue<int> Q;
        fill(prev, prev+MAX_V, -1);
        fill(dist, dist+MAX_V, INF);
        dist[S] = 0;
        in_Q[S] = true;
        Q.push(S);

        while(!Q.empty()){
            int curr = Q.front();
            Q.pop();
            in_Q[curr] = false;
            for (int next:graph[curr]){
                if (C[curr][next] - F[curr][next] > 0 && dist[next] >dist[curr] + D[curr][next]){
                    dist[next] = dist[curr] + D[curr][next];
                    prev[next] = curr;
                    if (!in_Q[next]){
                        Q.push(next);
                        in_Q[next] = true;
                    }
                }
            }
        }
```

```
78          for (int i=E;i!=S;i=prev[i]){
79              cost += D[prev[i]][i];
80              F[prev[i]][i] += 1;
81              F[i][prev[i]] -= 1;
82          }
83      }
84
85
86      printf("%d\n", cost);
87
88  }
89
```