#include <iostream>

#include <queue>

#include <cstring>

#define INF 1e9

using namespace std;

int N, M, X;

// pair<노드, 비용>

vector<pair<int,int>> v[1001];

int d[1001];

void dijkstra(int start, int dest){

priority\_queue<pair<int, int>> pq;

pq.push({0, start});

d[start] = 0;

while(!pq.empty()){

int dist = -pq.top().first;

int now = pq.top().second;

pq.pop();

if(now == dest){

break;

}

if(d[now] < dist)

continue;

for(int i=0; i<v[now].size(); i++){

int cost = dist + v[now][i].second;

if(cost < d[v[now][i].first]){

d[v[now][i].first] = cost;

pq.push({-cost, v[now][i].first});

}

}

}

}

int main(void){

cin >> N >> M >> X;

for(int i=0; i<M; i++){

int s, e, t;

cin >> s >> e >> t;

v[s].push\_back({e, t});

}

int ans = 0;

for(int i=1; i<=N; i++){

int tmp = 0;

fill\_n(d, 1001, INF);

dijkstra(X, i);

tmp += d[i];

fill\_n(d, 1001, INF);

dijkstra(i, X);

tmp += d[X];

if(ans < tmp)

ans = tmp;

}

cout << ans;

}