/\*

PROG: revamp

LANG: C++

ID: hayk.sa1

\*/

#include <cstdio>

#include <algorithm>

using namespace std;

#define inf 2000000000

int n, k;

int e[50010][3];

int \*g[10010], \*w[10010], deg[10010];

void init()

{

freopen("revamp.in", "r", stdin);

freopen("revamp.out", "w", stdout);

int m, i;

scanf("%d%d%d", &n, &m, &k);

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

{

scanf("%d%d%d", e[i], e[i]+1, e[i]+2);

deg[--e[i][0]]++;

deg[--e[i][1]]++;

}

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

{

g[i] = new int[deg[i]+1];

w[i] = new int[deg[i]+1];

deg[i] = 0;

}

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

{

g[e[i][0]][deg[e[i][0]]] = e[i][1];

w[e[i][0]][deg[e[i][0]]++] = e[i][2];

g[e[i][1]][deg[e[i][1]]] = e[i][0];

w[e[i][1]][deg[e[i][1]]++] = e[i][2];

}

}

int m;

int d[10010][21];

int pl[10010][21];

struct node

{

int p, q;

} h[500000];

void up(int p)

{

while (p)

{

node &u=h[p], &v=h[p>>1];

if (d[u.p][u.q] >= d[v.p][v.q])

return;

swap(u, v);

swap(pl[u.p][u.q], pl[v.p][v.q]);

p >>= 1;

}

}

void pop()

{

int p=0, q, l, r;

swap(h[0], h[m]);

swap(pl[h[0].p][h[0].q], pl[h[m].p][h[m].q]);

m--;

while (p < m)

{

q = p;

l = p<<1;

r = (p<<1)|1;

if (l < m && d[h[q].p][h[q].q] > d[h[l].p][h[l].q])

q = l;

if (r < m && d[h[q].p][h[q].q] > d[h[r].p][h[r].q])

q = r;

if (p == q)

return;

swap(h[p], h[q]);

swap(pl[h[p].p][h[p].q], pl[h[q].p][h[q].q]);

p = q;

}

}

int dijkstra()

{

int p, q, u, i, j;

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

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

{

if (i+j)

d[i][j] = inf;

h[m].p = i;

h[m].q = j;

pl[i][j] = m;

up(m++);

}

while (m)

{

p = h[0].p;

q = h[0].q;

if (p == n-1 && q == k)

return d[p][q];

pop();

for (i = deg[p]-1; i >= 0; i--)

{

u = g[p][i];

if (d[u][q] > d[p][q]+w[p][i])

{

d[u][q] = d[p][q]+w[p][i];

up(pl[u][q]);

}

}

if (q < k)

for (i = deg[p]-1; i >= 0; i--)

{

u = g[p][i];

if (d[u][q+1] > d[p][q])

{

d[u][q+1] = d[p][q];

up(pl[u][q+1]);

}

if (d[p][q+1] > d[p][q])

{

d[p][q+1] = d[p][q];

up(pl[p][q+1]);

}

}

}

return -1;

}

int main()

{

init();

printf("%d\n", dijkstra());

return 0;

}