#include

#include

#include

using namespace std;

int n, c;

int e[50010][3];

map m;

vector > g[10010];

int solve()

{

int ret = 0, i, j;

map::iterator I;

m.clear();

for (i = 1; i <= n; i++)

{

I = m.find(i);

if (I != m.end())

{

ret += m[i];

c += m[i];

m.erase(I);

}

for (j = g[i].size()-1; j >= 0; j--)

{

m[g[i][j].first] += g[i][j].second;

c -= g[i][j].second;

}

while (c < 0)

{

I = m.end();

--I;

if (c+I->second < 0)

{

c += I->second;

m.erase(I);

}

else

{

I->second += c;

if (I->second == 0)

m.erase(I);

c = 0;

}

}

g[i].clear();

}

return ret;

}

int main()

{

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

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

int p, k, i;

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

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

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

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

if (e[i][0] < e[i][1])

g[e[i][0]].push\_back(make\_pair(e[i][1], e[i][2]));

p = solve();

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

if (e[i][0] > e[i][1])

g[n-e[i][0]+1].push\_back(make\_pair(n-e[i][1]+1, e[i][2]));

printf("%d\n", p+solve());

return 0;

}