

```

1 // Problem: Machine Works
2 // Contest: Virtual Judge - UVA
3
4 //  $x*m1+c1 = x*m2+c2$ 
5 //  $x(m1-m2) = c2-c1$ 
6 //  $x=(c2-c1)/(m1-m2)$ 
7
8 double intersect(const pii &l1, const pii &l2){
9     return double(l2.second-l1.second)/double(l1.first-l2.first);
10 }
11
12 bool usefull(const pii &l1, const pii &l2, const pii &l3){
13     double d12 = intersect(l1, l2);
14     double d23 = intersect(l2, l3);
15     return d12 <= d23;
16 }
17
18 int calc(const pii &seg, int x){
19     return x * seg.first + seg.second;
20 }
21
22 int32_t main(){
23
24     fastIO;
25
26 #ifdef LOCAL
27     freopenI;
28     freopenO;
29 #endif
30
31     // freopen("name.in", "r", stdin);
32
33     int tt = 1;
34     int n, c, d;
35     while(cin >> n >> c >> d, n){
36
37         // d p r g
38
39         vi D(n), P(n), R(n), G(n);
40         for(int i = 0; i < n; i++){
41             cin >> D[i] >> P[i] >> R[i] >> G[i];
42         }
43
44         n++;
45         D.push_back(d+1);
46         P.push_back(0);
47         R.push_back(c);
48         G.push_back(0);
49
50         vi pos(n);
51         iota(pos.begin(), pos.end(), 0);
52         sort(pos.begin(), pos.end(), [&](int i, int j){
53             return D[i] < D[j];
54         });
55
56         map<int, int> mp;
57         mp[0] = c;
58         int cans;
59
60         for(int i = 0; i < n; i++){
61
62             int p = pos[i];
63             int x = D[p];
64
65             while(mp.size() > 1){
66
67                 auto f = mp.begin();
68                 auto s = mp.begin(); s++;
69
70                 if(calc(*f, x) <= calc(*s, x)){
71                     mp.erase(f);
72                 }
73                 else{

```

```

74         break;
75     }
76 }
77
78 cans = calc(*mp.begin(), x);
79 if(cans < P[p]) continue;
80
81 // dp[i] = dp[j] + arr[j].R - arr[j].P + arr[j].G * (arr[i].D - arr[j].D
82 -1);
83 // dp[i] = arr[i].D * arr[j].G + (dp[j] + arr[j].R - arr[j].P - arr[j].G
84 * (arr[j].D+1))
85 // x = arr[i].D
86 // m = arr[j].G
87 // c = (dp[j] + arr[j].R - arr[j].P - arr[j].G * (arr[j].D+1)
88
89 int m = G[p];
90 int c = cans + R[p] - P[p] - G[p] * (D[p]+1);
91
92 auto l2 = mp.emplace(m, c).first;
93 l2->second = max(l2->second, c);
94
95 if(l2 != mp.begin() && l2 != --mp.end()){
96
97     auto l1 = l2; l1--;
98     auto l3 = l2; l3++;
99
100     if(!usefull(*l1, *l2, *l3)){
101         mp.erase(l2);
102         continue;
103     }
104 }
105
106 while(mp.size() > 1 && l2 != mp.begin() && l2 != ++mp.begin()){
107
108     auto l1 = l2; l1--;
109     auto l0 = l1; l0--;
110
111     if(!usefull(*l0, *l1, *l2)){
112         mp.erase(l1);
113     }
114     else{
115         break;
116     }
117 }
118
119 cout << "Case " << tt++ << ": " << cans << endl;
120 }
121

```