

```

1 // [木塊問題/The Blocks Problem](1/3)
2 #define IN "P14IN.txt"
3 #define OUT "P14OUT.txt"
4 //*****
5 #include <iostream>
6 #include <ctime>
7 using namespace std;
8 void redir(void);
9 //*****
10 /* Work Space*/
11 #include <string>
12 #include <vector>
13
14 void find_block(int a, int &pa, int &ha);
15 void clear_above(int p, int h);
16 void pipe_over(int p, int h, int p2);
17 void show(void);
18
19 int n;
20 vector<int> pipe[25]; //二維陣列(第一維大小根據題意, 第二維大小不固定)
21 //*****
22 int main(void)
23 {
24     redir(); //redirection
25 //*****
26 /* Work Space*/
27     int i;
28     string s1, s2;
29     int a, b;
30     int pa, pb; //pa, pb: position of a and b
31     int ha, hb; //ha, hb: height of a and b
32
33     cin >> n;
34     for(i=0; i<n; i++){
35         pipe[i].push_back(i);
36     }
37
38     while(1){
39         cin >> s1;
40         if(s1 == "quit"){ // '=' is overloaded for type string
41             break;
42         }else{
43             cin >> a >> s2 >> b;
44
45             find_block(a, pa, ha);
46             find_block(b, pb, hb);
47
48             if(pa == pb){
49                 continue; //非法指令
50             }
51             if(s1 == "move"){
52                 clear_above(pa, ha);
53             }
54
55             if(s2 == "onto"){
56                 clear_above(pb, hb);
57             }
58
59             pipe_over(pa, ha, pb);
60         }
61     }
62     show();
63 //*****

```

```

64 //[木塊問題/The Blocks Problem](2/3)
65     freopen("CON", "r", stdin); //取消重新導向
66     freopen("CON", "w", stdout);
67
68     printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
69
70     system("pause");
71     return 0; //the end...
72 }
73
74 void redir(void)
75 {
76     freopen(IN, "r", stdin);
77     freopen(OUT, "w", stdout);
78 }
79 //*****
80 /* Work Space*/
81 //發現木塊a的位置p和高度h(皆從0算起)
82 void find_block(int a, int &p, int &h)
83 {
84     for(p=0; p<n; p++){
85         for(h=0; h<pipe[p].size(); h++){
86             if(pipe[p][h] == a){
87                 return;
88             }
89         }
90     }
91 }
92
93 //把第p堆高度為h的木塊"上方"的所有木塊歸位
94 void clear_above(int p, int h)
95 {
96     int i;
97     int b;
98
99     for(i=pipe[p].size()-1; i>h; i--){
100         b = pipe[p][i];
101         pipe[b].push_back(b);
102     }
103     pipe[p].resize(h+1);
104 }
105
106 //把第p堆高度h"及其上方"的木塊整體移動到第p2堆的頂部
107 void pipe_over(int p, int h, int p2)
108 {
109     int i;
110
111     for(i=h; i<pipe[p].size(); i++){
112         pipe[p2].push_back(pipe[p][i]);
113     }
114     pipe[p].resize(h);
115 }
116
117
118
119
120
121
122
123
124
125
126

```

```
127 //[木塊問題/The Blocks Problem](3/3)
128 //輸出結果
129 void show(void)
130 {
131     int i, j;
132
133     for(i=0; i<n; i++){
134         printf("%d: ", i);
135         for(j=0; j<pipe[i].size(); j++){
136             printf("%d ", pipe[i][j]);
137         }
138         printf("\n");
139     }
140 }
141
142 //Input(IN) Sample
143 /*
144 10
145 move 9 onto 1
146 move 8 over 1
147 move 7 over 1
148 move 6 over 1
149 pile 8 over 6
150 pile 8 over 5
151 move 2 over 1
152 move 4 over 9
153 quit
154 */
155
156 //Output(OUT)
157 /*
158 0: 0
159 1: 1 9 2 4
160 2:
161 3: 3
162 4:
163 5: 5 8 7 6
164 6:
165 7:
166 8:
167 9:
168 */
```