A 蜜蜂

```cpp
#include<stdio.h>
int main()
{
    long long f[60];
    int a,b,i,T;
    scanf("%d",&T);
    f[0]=0;
    while(T--)
    {
        scanf("%d%d",&a,&b);
        f[1]=1;
        f[2]=2;
        f[3]=3;
        for(i=4;i<=b-a;++i)
        {
            f[i]=f[i-2]+f[i-1];
        }
        printf("%lld\n",f[b-a]);
    }
    return 0;
}
```

B 要变多少次

```cpp
#include <iostream>
#include <cstring>
#include <algorithm>
#include <fstream>
#define MAX 1001
using namespace std;

int main()
{
  int t;
  char s[MAX];
  //ifstream cin("test.txt");
  cin >> t;
  while (t--)
  {
    int l, cnt0, cnt1;
    int ans = 1000000;

    cin >> s;

    l = strlen(s);
    for (int i = 0; i <= l; i++)
    {
      cnt0 = cnt1 = 0;
      for (int j = i - 1; j >= 0; j--)
      {
        cnt0 += (s[j] == '0' ? 0 : 1);
      }
      for (int j = i; j < l; j++)
      {
        cnt1 += (s[j] == '1' ? 0 : 1);
      }
```

```
      ans = min(ans, cnt1 + cnt0);

    }
    cout << ans << endl;
  }
  //getchar();
  return 0;
}
```

C 未名冰场

```
#include <cstdio>
#include <iostream>
#include <queue>
#include <algorithm>
#include <cstring>
#include <cstdlib>

using namespace std;

int N,M;
int v[111][111],a[111][111];
const int dx[8] = {0,0,1,-1,1,1,-1,-1};
const int dy[8] = {1,-1,0,0,1,-1,1,-1};

int bfs(int x, int y) {
  queue<pair<int,int> > Q;
  Q.push(make_pair(x,y));
  v[x][y] = 1;
  while (!Q.empty()) {
    pair<int, int> pos = Q.front();Q.pop();
    for (int k = 0; k < 8; k++) {
      int xx = pos.first + dx[k];
      int yy = pos.second + dy[k];
      if (xx < 0 || yy < 0 || xx >= N || yy >= M) continue;
      if (a[xx][yy] == 0 || v[xx][yy] == 1) continue;
      v[xx][yy] = 1;
      Q.push(make_pair(xx,yy));
    }
  }return 1;
}

int main() {
  while (scanf("%d %d\n",&N,&M)!=EOF) {
    if (M == 0) break;
    for (int i = 0; i < N; i++) {
      for (int j = 0; j < M; j++) {
        char c;
        scanf("%c",&c);
        if (c == '*') a[i][j] = 0;
        else a[i][j] = 1;
      }scanf("\n");
    }
    int ans = 0;
    memset(v,0,sizeof(v));
    for (int i = 0; i < N; i++)
      for (int j = 0; j < M; j++)
        if (v[i][j] == 0 && a[i][j] == 1)
```

```
48          if (bfs(i,j)) ans++;
49      printf("%d\n",ans);
50    }
51  }
```

D 迷阵

```
1   #include <iostream>
2   #include <cstring>
3   #include <queue>
4   using namespace std;
5
6   struct state
7   {
8       int hp;
9       int step;
10      int x,y;
11      state(int _x=0, int _y=0, int _hp=0, int _step=0):x(_x),y(_y),hp(_hp),step(_step){}
12  };
13
14
15  const int MAXN = 255;
16  char MAP[MAXN][MAXN];
17  int visit[MAXN][MAXN];
18
19  int dirs[4][2] = {{0,1},{0,-1},{-1,0},{1,0}};
20
21  int main()
22  {
23      int T;
24      cin >> T;
25      while(T-->0)
26      {
27          memset(MAP, 0, sizeof(MAP));
28          memset(visit, 0, sizeof(visit));
29          int m,n,h;
30          cin >> m >> n >> h;
31          for (int i = 0; i < m; ++i)
32          {
33              for (int j = 0; j < n; ++j)
34              {
35                  cin >> MAP[i][j];
36              }
37          }
38
39          queue<state> q;
40          q.push(state(0,0,h));
41          visit[0][0]=h;
42          int flag = 0;
43          while(!q.empty())
44          {
45              state crt = q.front();
46              q.pop();
47              for (int i = 0; i < 4; ++i)
48              {
49                  int dx = dirs[i][0], dy = dirs[i][1];
50                  int nx = crt.x + dx; int ny = crt.y + dy;
51
```

```
                if (nx == m-1 && ny == n-1)
                {
                    cout<<crt.step+1<<endl;
                    flag = 1;
                    break;
                }

                if(nx >= 0 && ny >= 0)
                    if (nx < m && ny < n)
                    {
                        if(MAP[nx][ny] == '.' && visit[nx][ny] < crt.hp)
                        {
                            q.push(state(nx, ny, crt.hp, crt.step+1));
                            visit[nx][ny] = crt.hp;
                        }
                        else if(MAP[nx][ny] == '*' && crt.hp - 1 >= 0 && visit[nx][ny] < crt.hp - 1)
                        {
                            q.push(state(nx, ny, crt.hp-1, crt.step+1));
                            visit[nx][ny] = crt.hp-1;
                        }
                    }
            }
            if(flag) break;
        }
    }
}
```

E 表达式的期望值

```
#include <stdio.h>

#define MAXN 205

int n;
int     Ai[MAXN];
char    Oi[MAXN][6];
double  Pi[MAXN];

double ans;
int main()
{
    int i, j, ncase = 0;
    double q;
    while(scanf("%d", &n) == 1)
    {
        ncase++;

        for(i = 0; i <= n; i++)
            scanf("%d", &Ai[i]);

        for(i = 1; i <= n; i++)
            scanf("%s", Oi[i]);

        for(i = 1; i <= n; i++)
            scanf("%lf", &Pi[i]);

        ans = 0;
        // for each bit [0, 20]
```

```
        for(i = 0; i <= 20; i++)
        {
            q = 0;
            if(Ai[0] & (1 << i))
                q = 1;

            for(j = 1; j <= n; j++)
            {
                if(Ai[j] & (1 << i))
                {
                   if(Oi[j][0] == '&')
                       continue;
                   else if (Oi[j][0] == '^')
                       q = q * Pi[j] +  (1 - q) * (1 - Pi[j]);
                   else if (Oi[j][0] == '|')
                       q = q + (1 - q) * (1 - Pi[j]);
                }else{
                  if(Oi[j][0] == '&')
                      q = q *  Pi[j];
                }

            }

            ans += q * (1 << i);
        }

        printf("Case %d:\n%.6lf\n", ncase, ans);
    }
}
```

F 数字变换

```
#include<stdio.h>
#include<queue>

using namespace std;

int vis[100000][4][3];

struct Move {
  int num, move, second, third;
  Move(int _num, int _move, int _second, int _third):num(_num),move(_move),second(_second),third(_third){}
};

void int2char(int x, char c[5]) {
  for (int i = 0; i < 5; ++i) {
    c[i] = '0' + x % 10;
    x /= 10;
  }
}

int char2int(char c[5]) {
  int sum = 0;
  for (int i = 4; i >= 0; --i) {
    sum *= 10;
    sum += c[i] - '0';
  }
  return sum;
```

```
}

int swap(int x, int pos) {
  char s[5];
  int2char(x, s);
  char t = s[pos];
  s[pos] = s[pos + 1];
  s[pos + 1] = t;
  return char2int(s);
}

int add1(int x, int pos) {
  char s[5];
  int2char(x, s);
  s[pos] = (s[pos] - '0' + 1) % 10 + '0';
  return char2int(s);
}

int mul2(int x, int pos) {
  char s[5];
  int2char(x, s);
  s[pos] = (s[pos] - '0') * 2 % 10 + '0';
  return char2int(s);
}

int main() {
  for (int i = 0; i < 100000; ++i)
    for (int j = 0; j < 4; ++j)
      for (int k = 0; k < 3; ++k)
        vis[i][j][k] = -1;
  vis[12345][0][0] = 0;
  queue<Move> q;
  q.push(Move(12345, 0, 0, 0));
  while(!q.empty()) {
    Move m = q.front();
    q.pop();
    int num = m.num, move = m.move, sec = m.second, thr = m.third;
    for (int i = 0; i < 4; ++i) {
      int s = swap(num, i);
      if (vis[s][sec][thr] == -1) {
        vis[s][sec][thr] = move + 1;
        q.push(Move(s, move + 1, sec, thr));
      }
    }
    if (sec < 3)
      for (int i = 0; i < 5; ++i) {
        int s = add1(num, i);
        if (vis[s][sec + 1][thr] == -1) {
          vis[s][sec + 1][thr] = move + 1;
          q.push(Move(s, move + 1, sec + 1, thr));
        }
      }
    if (thr < 2)
      for (int i = 0; i < 5; ++i) {
        int s = mul2(num, i);
        if (vis[s][sec][thr + 1] == -1) {
          vis[s][sec][thr + 1] = move + 1;
          q.push(Move(s, move + 1, sec, thr + 1));
```

```
        }
      }
  }
  char s[6], c[5];
  while(scanf("%s", s) != EOF) {
    for (int i = 0; i < 5; ++i) c[i] = s[4 - i];
    int ans = -1;
    int num = char2int(c);
    for (int i = 0; i < 4; ++i)
      for (int j = 0; j < 3; ++j)
        if (ans == -1 || vis[num][i][j] < ans && vis[num][i][j] != -1) ans = vis[num][i][j];
    printf("%d\n", ans);
  }
}
```

G 忍者道具

```
#include <iostream>
#include <fstream>
#include <algorithm>
#include <vector>

using namespace std;

int find(const int size, int curSize, const vector<int>& v, int last){
  int index = -1;
  if (last == 0){
    return -1;
  }

  if (v[last - 1] + curSize <= size){
    return last - 1;
  }

  if (last == 1){
    return -1;
  }

  for (int i = last - 2; i--; i >= 0){
    if (curSize + v[i] <= size && curSize + v[i + 1] > size){
      index = i;
      break;
    }
  }

  return index;
}

int main(){
  int n = 0;
  int w = 0;
  cin >> n;
  cin >> w;
  vector<int> tools;
  int sum = 0;
  for (int i = 0; i < n; i++){
    int a = 0;
    cin >> a;
```

```
    tools.push_back(a);
    sum += a;
  }

  sort(tools.begin(), tools.end());

  int minBags = sum / w;
  if (sum%w != 0){
    minBags++;
  }

  int maxBags = n;

  for (int nBags = minBags; nBags <= maxBags; nBags++){
    int nUsed = 0;
    while (nUsed <= nBags){
      int curSize = 0;
      curSize += tools[tools.size() - 1];


      int last = tools.size();
      vector<int> indeces;
      while (curSize <= w){
        int index = find(w, curSize, tools, last);
        if (index == -1){
          nUsed++;
          break;
        }
        else{
          indeces.push_back(index);
          curSize += tools[index];
          last = index;
        }
      }

      if (indeces.size() > 0){
        for (int i = 0; i < indeces.size(); i++){
          tools.erase(tools.begin() + indeces[i]);
        }
      }

      if (tools.size() == 0){
        break;
      }

    }

    if (tools.size() == 0){
      cout << nBags << endl;
      break;
    }
  }

  return 0;
}
```

H 构建世界

```cpp
#include<cstdio>
#include<cstring>
#include<cctype>
#include<algorithm>
#define r(a,b) ((a)*W+(b))
using namespace std;

int Next(int a,int b){
  int ret=a;
  while (ret<b) ret+=a;
  return ret;
}

int gcd(int a,int b){return b?gcd(b,a%b):a;}
int ggg(int a,int b){return a/gcd(a,b)*b;}

void get(char *s,int &tot,char &fh){
  tot=1;int a=0,b=0;
  for (;*s;++s){
    if (*s=='(') ++a;
    else if (*s==')') {
      if (--a<0) return;
    }
    else if (*s=='[') {
      ++b;
      if (a==0) ++tot;
    }
    else if (*s==']') --b;
    else if (*s==':' && a==0) fh=':',++tot;
    else if (*s=='.' && a==0) fh='.',++tot;
  }
}

void calc(char*&s,int&minh,int &gh,int &minw,int &gw){
  if (*s=='('){
    char fh;
    minh=minw=0;gh=gw=1;
    int tot=0;
    get(s+1,tot,fh);
    do{
      int tminh,tgh,tminw,tgw,bt;
      ++s;
      if (*s=='['){
        ++s;
        calc(s,tminh,tgh,tminw,tgw);
        ++s;
        if (fh==':'){
          tminh=tminh/2+tminh%2;
          if (tgh%2==0) tgh/=2;
        }
        else if (fh=='.'){
          tminw=tminw/2+tminw%2;
          if (tgw%2==0) tgw/=2;
        }
      }
      else{
        calc(s,tminh,tgh,tminw,tgw);
```

```
58        }
59        minh=max(tminh,minh);
60        minw=max(tminw,minw);
61        gh=ggg(gh,tgh);
62        gw=ggg(gw,tgw);
63      }while (*s!=')');
64      if (fh==':') minh*=tot,gh*=tot;
65      else minw*=tot,gw*=tot;
66      ++s;
67    }
68    else {
69      int len=0;
70      while (isdigit(*s)) ++len,++s;
71      minh=2;
72      minw=len+3;
73      gh=1;
74      gw=1;
75    }
76  }
77
78  char ans[200000];
79  int W;
80
81  void print(char*&s,int x,int y,int h,int w){
82    int i;
83    for (i=0;i<=w;++i){
84      if (ans[r(x,y+i)]==' ') ans[r(x,y+i)]='-';
85      if (ans[r(x+h,y+i)]==' ') ans[r(x+h,y+i)]='-';
86    }
87    for (i=0;i<=h;++i){
88      if (ans[r(x+i,y)]==' ') ans[r(x+i,y)]='|';
89      if (ans[r(x+i,y+w)]==' ') ans[r(x+i,y+w)]='|';
90    }
91    ans[r(x,y)]=ans[r(x,y+w)]=ans[r(x+h,y)]=ans[r(x+h,y+w)]='+';
92    if (*s=='('){
93      char fh;
94      int tot=0,tmp=0,nx,ny,nh,nw;
95      get(s+1,tot,fh);
96      do{
97        ++s;
98        if (*s=='['){
99          ++s;
100         if (fh==':'){
101           nx=x+h/tot*tmp;
102           ny=y;
103           nh=h/tot*2;
104           nw=w;
105         }
106         else{
107           nx=x;
108           ny=y+w/tot*tmp;
109           nh=h;
110           nw=w/tot*2;
111         }
112         tmp+=2;
113         print(s,nx,ny,nh,nw);
114         ++s;
115       }
```

```
      else{
        if (fh==':'){
          nx=x+h/tot*tmp;
          ny=y;
          nh=h/tot;
          nw=w;
        }
        else{
          nx=x;
          ny=y+w/tot*tmp;
          nh=h;
          nw=w/tot;
        }
        tmp++;
        print(s,nx,ny,nh,nw);
      }
    }while (*s!=')');
    ++s;
  }
  else {
    int len=0;
    y+=2;
    while (isdigit(*s)) ans[r(x+1,y++)]=*s,++len,++s;
  }
}


char s[10000];
int minh,gh,minw,gw,h,w;

int main()
{
  fgets(s, sizeof s, stdin);
  char *S=s;
  calc(S,minh,gh,minw,gw);
  h=Next(gh,minh);w=Next(gw,minw);
  memset(ans,' ',sizeof ans);
  W=w+2;
  S=s;
  print(S,0,0,h,w);
  int i,j;
  for (i=0;i<=h;++i,putchar('\n'))
    for (j=0;j<=w;++j)
      putchar(ans[r(i,j)]);
}
```

I 课程大作业

```
#include <iostream>
#include <cstring>
#include <string>
#include <algorithm>
#include <vector>
using namespace std;


int t;
int n;
```

```
11  struct Course
12  {
13    string name;
14    int d;
15    int c;
16  };
17  Course cs[20];
18
19
20  struct Node
21  {
22    int pre;
23    int minScore;
24    int last;
25    int finishDay;
26  };
27
28  Node dp[(1 << 16) + 20];
29  vector<int> GetPath( int status)
30  {
31    vector<int> path;
32    while( status  ) {
33      path.push_back(dp[status].last);
34      status = dp[status].pre;
35    }
36    reverse(path.begin(),path.end());
37    return path;
38  }
39  int main()
40  {
41    cin >> t;
42    while(t--) {
43      cin >> n;
44      char name[60];
45      int d,c;
46      for(int i = 0; i < n; ++i)
47        cin >> cs[i].name >> cs[i].d >> cs[i].c;
48      dp[0].finishDay = 0;
49      dp[0].minScore = 0;
50      dp[0].pre = -1;
51      int m = 1 << n;
52      for(int i = 1; i < m; ++i) {
53        dp[i].minScore = 1 << 30;
54        for(int j = 0; j < n; ++j) {
55
56          if( i & ( 1 << j )) {
57            int pre = i - ( 1 << j);
58
59            int finishDay = dp[pre].finishDay + cs[j].c;
60            int tmpScore = finishDay - cs[j].d;
61            if( tmpScore < 0)
62              tmpScore = 0;
63            if( dp[i].minScore > dp[pre].minScore + tmpScore ) {
64              dp[i].minScore = dp[pre].minScore + tmpScore ;
65              dp[i].pre = pre;
66              dp[i].finishDay = finishDay;
67              dp[i].last = j;
68            }
```

```
          if( dp[i].minScore == dp[pre].minScore + tmpScore ) {
            vector<int> p1 = GetPath(dp[i].pre);
            vector<int> p2 =  GetPath(pre);
            if ( p2 < p1) {
              dp[i].pre = pre;
              dp[i].finishDay = finishDay;
              dp[i].last = j;
            }
          }
        }
      }
    }
    cout << dp[m-1].minScore << endl;

    int status = m-1;
    vector<int> path = GetPath(status);

    for( int i = 0; i < path.size(); ++ i)
      cout << cs[path[i]].name << endl;
  }
  return 0;
}
```

J ACM ICPC 2018

```
/*
    Two-pass Multi-Knapsack problem.
    1) In the first phase, we select the cakes to minimize the space under energy constraints.
    2) In the second phase, we select trunks to minimize the cost under space constraints
*/

#include <stdio.h>
#include <memory.h>
#define INF 0x3f3f3f3f
#define MAXN 205 * 205
#define MAXV 205
#define MAX_VAL 60010
int weight[MAXN], value[MAXN];
int num;
int ans[MAX_VAL];

int min(int a, int b)
{
    if (a < b) return a;
    return b;
}

int max(int a, int b)
{
    if(a > b) return a;
    return b;
}

int binary(int s, int * a, int * b, int * c)
{
     int i, t, sum;
     num = 0;
     sum = 0;
```

```
34      for(i = 0 ; i < s; i++)
35      {
36         t = 1;
37         sum += c[i] * b[i];
38         while(c[i] >= t)
39         {
40               weight[num] = a[i] * t;
41               value[num]  = b[i] * t;
42               c[i]        = c[i] - t;
43               t = 2 * t;
44               num++;
45         }
46         if(c[i] > 0)
47         {
48            weight[num] = a[i] * c[i]; value[num] = b[i] * c[i]; num++;
49         }
50      }
51      return sum;
52  }
53
54  int main()
55  {
56      int i, j, n, k, temp, ntest;
57      int num_cake, num_truck, num_energy;
58      int min_space, min_cost, res;
59      int a[MAXV],b[MAXV],c[MAXV];
60
61      scanf("%d", &ntest);
62      for(n = 0; n < ntest; n++)
63      {
64          scanf("%d%d%d", &num_cake, &num_truck, &num_energy);
65          for(k = 0; k < num_cake; k++)
66               scanf("%d%d%d", &a[k], &b[k], &c[k]);
67
68          temp = binary(num_cake, a, b, c);
69
70          //for(k = 0; k < num; k++)
71            //   printf("%d %d\n", weight[k], value[k]);
72
73          /* 1st phase: multi-knapsack problem.
74             ans[i]: the mimimum size of the disserts given >=i energy */
75          min_space = INF;
76          memset(ans, INF, sizeof(ans));
77
78          ans[0] = 0;
79          for(i = 0; i < num; i++)
80             for(j = MAX_VAL - 1; j >= weight[i]; j--)
81             {
82                 ans[j] = min(ans[j], ans[j - weight[i]] + value[i]);
83                 if(j >= num_energy)
84                 {
85                     if(min_space > ans[j])
86                     {
87                        min_space = ans[j];
88                     }
89                 }
90             }
91
```

```
92
93          //printf("min_space %d\n", min_space);
94
95          for(k = 0; k < num_truck; k++)
96              scanf("%d%d%d", &b[k], &a[k], &c[k]);
97
98          temp = binary(num_truck, a, b, c);
99          if(temp < min_space)
100         {
101            printf("FAIL\n");
102            continue;
103         }
104         //for(k = 0; k < num; k++)
105           //   printf("%d %d\n", weight[k], value[k]);
106
107         /* 2nd phase: multi-knapsack problem.
108            ans[i]: the maximum volume achieved with cost i.
109         */
110
111         res = INF;
112         memset(ans, 0, sizeof(ans));
113         for(i = 0; i < num; i++)
114             for(j = 50000; j >= weight[i]; j--)
115             {
116                 ans[j] = max(ans[j], ans[j - weight[i]] + value[i]);
117                 if(ans[j] >= min_space)
118                       res = min(res, j);
119             }
120         if(res <= 50000)
121             printf("%d\n", res);
122         else
123             printf("FAIL\n");
124
125     }
126
127     return 0;
128 }
129
130
131
```

K 机智的弗兰克

```
1  #include <iostream>
2  using namespace std;
3
4  int ans;
5  int N;
6  int t;
7
8  void solve() {
9    //cout <<"Test Case: " << t << endl;
10   t++;
11   int n, m, L, l, r, t, b;
12   cin >> n >> m >> L;
13   cin >> l >> r >> b >> t;
14   if (L >= 2 * (m + n)) {
15     cout << m*n - (r - l)*(t - b) << endl;
```

```
    return;
  }
  int l1, r1, t1, b1;
  l1 = l;
  r1 = r;
  t1 = t;
  b1 = b;
  if (l + r <= m) {
    l1 = m - r;
    r1 = m - l;
  }
  if (t + b <= n) {
    t1 = n - b;
    b1 = n - t;
  }
  int ans = 0;
  for (int i = 1; i < L / 2 && i <= m; i++) {
    for (int j = 1; j <= L / 2 - i && j <= n; j++) {
      if (i <= l1 || j <= b1)
        ans = ans > i*j ? ans : i*j;
      if (i > l1 && i <= r1 && j > b1 && j <= t1)
        ans = ans > i*j - (i - l1)*(j - b1) ? ans : i*j - (i - l1)*(j - b1);
      if (i > l1 && i <= r1 && j >= t1)
        ans = ans > i*j - (i - l1)*(j - b1) ? ans : i*j - (i - l1)*(j - b1);
      if (i >=  r1 && j > b1 && j <= t1)
        ans = ans > i*j - (i - l1)*(j - b1) ? ans : i*j - (i - l1)*(j - b1);
      if (i >= r1 - l1 && j >= t1 - b1)
        ans = ans > i*j - (r - l)*(t - b) ? ans : i*j - (r - l)*(t - b);
    }
  }

  cout << ans << endl;
}

int main()
{
  t = 1;
  cin>>N;
  while(t<=N){
    solve();
  }
  return 0;
}
```

L 华容道

```
#include <stdio.h>
#include <iostream>
#include <unordered_map>
#include <unordered_set>
#include <string>
#include <vector>
#include <queue>
using namespace std;

typedef pair<int, int> Point;

unordered_set<string> flag;
```

```
13  unordered_map<int, bool> valid[3];
14
15  const int fx[4][2] = {{0, 1}, {1, 0}, {0, -1}, {-1, 0}};
16
17  #define MUL (1000)
18
19  class block {
20  public:
21    vector<Point> s;
22    int n, l, r, u, d;
23    bool map[10][10];
24
25    void read() {
26      Point tmp;
27      s.clear();
28      for (int i = 0;i < 10;++i)
29        for (int j = 0;j < 10;++j)
30          map[i][j] = false;
31      l = u = 9;
32      d = r = 0;
33      for (int i = 0;i < n;++i) {
34        scanf("%d%d", &tmp.first, &tmp.second);
35        map[tmp.first][tmp.second] = true;
36        s.push_back(tmp);
37        l = min(l, tmp.first); r = max(r, tmp.first);
38        u = min(u, tmp.second); d = max(d, tmp.second);
39      }
40    }
41
42    void print() {
43      cout << "lrud" << l << ' ' << ' ' << r << ' ' << u <<' ' << d << endl;
44    }
45  };
46
47  class State {
48  public:
49    int step, x1, x2, y1, y2;
50
51    string ToString() {
52      return to_string(x1) + '/' + to_string(x2) + '/' + to_string(y1) + '/' + to_string(y2);
53    }
54  };
55
56  block a[3];
57  queue<State> q;
58
59  bool if_block_ok(block &a, block &b, int dx, int dy) {
60    int ml, mu, mr, md;
61
62    dx = a.l + dx - b.l;
63    dy = a.u + dy - b.u;
64
65    ml = max(a.l, b.l + dx);
66    mu = max(a.u, b.u + dy);
67    mr = min(a.r, b.r + dx);
68    md = min(a.d, b.d + dy);
69
70    bool ans = (ml > mr || mu > md);
```

```
71
72    // a.print();
73    // b.print();
74
75    // cout << dx << ' ' << dy <<' ' << ans << endl;
76
77    return ans;
78  }
79
80  bool if_ok(int x1, int y1, int x2, int y2) {
81    if (if_block_ok(a[0], a[1], x1, y1) && if_block_ok(a[0], a[2], x2, y2)
82      && if_block_ok(a[1], a[2], x2 - x1, y2 - y1)) return true;
83
84    return false;
85  }
86
87  int code(int dx, int dy) {
88    return MUL * dx + dy;
89  }
90
91  bool pair_valid(int id, block &a, block &b, int dx, int dy) {
92    int c = code(dx, dy);
93    bool fl = true;
94    int x, y;
95
96    if (dx < -20 || dx > 20 || dy < -20 || dy > 20) return false;
97
98    //printf("s%d %d\n",a.n, b.n);
99
100   dx = a.l + dx - b.l;
101   dy = a.u + dy - b.u;
102   if (valid[id].find(c) == valid[id].end()) {
103     for (int i = 0;i < b.n;++i) {
104       x = b.s[i].first + dx;
105       y = b.s[i].second + dy;
106       //printf("a%d %d\n", x, y);
107       if (x >= 0 && x < 10 && y >= 0 && y < 10 && a.map[x][y]) {
108         fl = false;
109         break;
110       }
111     }
112     valid[id][c] = fl;
113   }
114
115   return valid[id][c];
116 }
117
118 bool if_valid(int x1, int y1, int x2, int y2) {
119   if (pair_valid(0, a[0], a[1], x1, y1) && pair_valid(1, a[0], a[2], x2, y2)
120     && pair_valid(2, a[1], a[2], x2 - x1, y2 - y1)) return true;
121
122   return false;
123 }
124
125
126 int bfs() {
127   State s, t;
128
```

```
129     s.step = 0;
130     s.x1 = a[1].l - a[0].l; s.y1 = a[1].u - a[0].u;
131     s.x2 = a[2].l - a[0].l; s.y2 = a[2].u - a[0].u;
132
133     //cout << "test"  << pair_valid(0, a[0], a[1], 2, 0) << endl;
134
135
136     if (if_ok(s.x1, s.y1, s.x2, s.y2)) return 0;
137     if (!if_valid(s.x1, s.y1, s.x2, s.y2)) return -1;
138     flag.insert(s.ToString());
139     q.push(s);
140
141
142
143     //return -1;
144
145     while (!q.empty()) {
146       s = q.front();
147       q.pop();
148
149       t.step = s.step + 1;
150       //cout << t.step << endl;
151
152       //move block 1
153       for (int i = 0;i < 4;++i) {
154         t.x1 = s.x1 + fx[i][0];
155         t.y1 = s.y1 + fx[i][1];
156         t.x2 = s.x2 + fx[i][0];
157         t.y2 = s.y2 + fx[i][1];
158
159         if (if_valid(t.x1, t.y1, t.x2, t.y2) && flag.find(t.ToString()) == flag.end()) {
160           if (if_ok(t.x1, t.y1, t.x2, t.y2)) {
161             //printf("%d %d %d %d\n", t.x1, t.y1, t.x2, t.y2);
162             return t.step;
163           }
164           flag.insert(t.ToString());
165           q.push(t);
166         }
167       }
168       //move block 2
169       for (int i = 0;i < 4;++i) {
170         t.x1 = s.x1 + fx[i][0];
171         t.y1 = s.y1 + fx[i][1];
172         t.x2 = s.x2;
173         t.y2 = s.y2;
174
175         if (if_valid(t.x1, t.y1, t.x2, t.y2) && flag.find(t.ToString()) == flag.end()) {
176           if (if_ok(t.x1, t.y1, t.x2, t.y2)) return t.step;
177           flag.insert(t.ToString());
178           q.push(t);
179         }
180       }
181       //move block 3
182       for (int i = 0;i < 4;++i) {
183         t.x1 = s.x1;
184         t.y1 = s.y1;
185         t.x2 = s.x2 + fx[i][0];
186         t.y2 = s.y2 + fx[i][1];
```

```
187
188         if (if_valid(t.x1, t.y1, t.x2, t.y2) && flag.find(t.ToString()) == flag.end()) {
189           if (if_ok(t.x1, t.y1, t.x2, t.y2)) return t.step;
190           flag.insert(t.ToString());
191           q.push(t);
192         }
193       }
194     }
195
196     return -1;
197   }
198
199   int main() {
200     //freopen("unlock.in", "r", stdin);
201     //freopen("output.txt", "w", stdout);
202     while (true) {
203       cin >> a[0].n >> a[1].n >> a[2].n;
204       if (a[0].n == 0 && a[1].n == 0 && a[2].n == 0) break;
205       for (int i = 0;i < 3;++i) {
206         a[i].read();
207         valid[i].clear();
208       }
209       if (a[0].n < a[1].n) swap(a[0], a[1]);
210       if (a[0].n < a[2].n) swap(a[0], a[2]);
211       if (a[1].n < a[2].n) swap(a[1], a[2]);
212
213       flag.clear();
214       while (!q.empty()) q.pop();
215       cout << bfs() << endl;
216     }
217   }
```