

```

1 int a[1001][1001],v[1001][1001],p[1001][1001];
2 int rowtree[1001][4001],coltree[1001][4001];
3 void rowUpdate(int id,int nd,int b,int e,int x,int val){
4     if(b==x&&e==x){ rowtree[id][nd] = val; return; }
5     int lf=2*nd, rg = 2*nd+1, md = (b+e)/2;
6     if(x<=md)rowUpdate(id,lf,b,md,x,val);
7     else rowUpdate(id,rg,md+1,e,x,val);
8     rowtree[id][nd] = max(rowtree[id][lf],rowtree[id][rg]);
9 }
10 void colUpdate(int id,int nd,int b,int e,int x,int val){
11     if(b==x&&e==x){ coltree[id][nd] = val; return; }
12     int lf=2*nd, rg = 2*nd+1, md = (b+e)/2;
13     if(x<=md)colUpdate(id,lf,b,md,x,val);
14     else colUpdate(id,rg,md+1,e,x,val);
15     coltree[id][nd] = max(coltree[id][lf],coltree[id][rg]);
16 }
17 int rowQuery(int id,int nd,int b,int e,int x,int y){
18     if(b>y||e<x)return -1000000000;
19     if(b>=x&&e<=y)return rowtree[id][nd];
20     int lf=2*nd, rg = 2*nd+1, md = (b+e)/2;
21     int m1 = rowQuery(id,lf,b,md,x,y);
22     int m2 = rowQuery(id,rg,md+1,e,x,y);
23     return max(m1,m2);
24 }
25 int colQuery(int id,int nd,int b,int e,int x,int y){
26     if(b>y||e<x)return -1000000000;
27     if(b>=x&&e<=y)return coltree[id][nd];
28     int lf=2*nd, rg = 2*nd+1, md = (b+e)/2;
29     int m1 = colQuery(id,lf,b,md,x,y);
30     int m2 = colQuery(id,rg,md+1,e,x,y);
31     return max(m1,m2);
32 }
33 int main(){
34     int tt; scanf("%d",&tt);
35     for(int ks=1; ks<=tt; ks++){
36         int n,m; scanf("%d%d",&n,&m);
37
38         for(int i=1; i<=n; i++)
39             for(int j=1; j<=m; j++)
40                 scanf("%d",&p[i][j]);
41
42         for(int i=1; i<=n; i++)
43             for(int j=1; j<=m; j++)
44                 scanf("%d",&v[i][j]);
45
46         for(int i=1; i<=1000; i++)
47             for(int j=1; j<=4000; j++)
48                 rowtree[i][j] = coltree[i][j] = -1000000000;
49
50         rowUpdate(n,1,1,m,m,v[n][m]);
51         colUpdate(m,1,1,n,n,v[n][m]);
52         a[n][m] = v[n][m];
53
54         for(int j=m; j>=1; j--){
55             for(int i=n; i>=1; i--){
56                 if(i==n&&j==m)continue;
57                 int rowcnt=-1000000000; int colcnt=-1000000000;
58                 if(j<m){
59                     int x = j+1; int y = min(j+p[i][j],m);
60                     rowcnt = rowQuery(i,1,1,m,x,y);
61                 }
62                 if(i<n){
63                     int x = i+1; int y = min(i+p[i][j],n);
64                     colcnt = colQuery(j,1,1,n,x,y);
65                 }
66                 int maxx = max(rowcnt,colcnt);
67                 a[i][j] = maxx+v[i][j];
68                 rowUpdate(i,1,1,m,j,maxx+v[i][j]);
69                 colUpdate(j,1,1,n,i,maxx+v[i][j]);
70             }
71         }
72         //int ans = rowQuery(1,1,1,m,1,1);
73         int ans = a[1][1];
74         printf("Case %d: %d\n",ks,ans);
75     }
76 }
```