

```

1  /** 10827 - Maximum sum on a torus (2D Maximum Sum):
2  A grid that wraps both horizontally and vertically is called a torus. Given a torus
3  where each cell contains an integer, determine the sub-rectangle with the largest sum.
4  The sum of a sub-rectangle is the sum of all the elements in that rectangle.
5
6  The first line in the input contains the number of test cases (at most 18).
7  Each case starts with an integer N (1<=N<=75) specifying the size of the torus
8  (always square). Then follows N lines describing the torus,
9  each line containing N integers between -100 and 100, inclusive.
10
11 For each test case, output a line containing a single integer:
12 the maximum sum of a sub-rectangle within the torus.
13 ****/
14 int n,row,col,maxx; int a[80][80],b[155][155],c[155][155],v[155],kd[155];
15 int fun(){
16     int rw=row+row; int cl=col+col;
17     for(int i=1,x=1; i<=rw; i++,x++){
18         if(x>row)x=1;
19         for(int j=1,y=1; j<=cl; j++,y++){
20             if(y>col) y=1;
21             b[i][j] = a[x][y];
22         }
23     }
24
25     for(int i=1; i<=cl; i++){
26         for(int j=1; j<=rw; j++){
27             c[i][j] = c[i][j-1] + b[j][i]; // be careful for b[j][i]
28         }
29     }
30
31     int ans=maxx;
32     for(int i=1; i<=row; i++){
33         for(int j=i; j<=row+i-1; j++){
34             for(int k=1; k<=cl; k++){
35                 v[k] = c[k][j]-c[k][i-1];
36             }
37             for(int x=1, kk=0; x<=cl; x++){ /// kadane or 1D maximum sum
38                 kd[x]=kd[x-1]+v[x]; kk++;
39
40                 if(kd[x]<0){
41                     kd[x]=0; kk=0;
42                 }else{
43                     ans = max(ans,kd[x]);
44                     if(kk==col){
45                         kk--; int p = x-col+1;
46                         kd[x] -= v[p]; p++;
47                         int mx=kd[x]; int mp=p;
48                         while(p<=x){
49                             kd[x] -= v[p]; p++; kk--;
50                             if(kd[x]>=mx){
51                                 mx=kd[x]; mp=p;
52                             }
53                         }
54                         kd[x]=mx; kk = (x-mp)+1;
55                     }
56                 }
57             }
58         }
59     }
60     return ans;
61 }
62 int main(){
63     int tt; scanf("%d",&tt);
64     while(tt--){
65         scanf("%d",&n);
66         row=col=n; maxx=-100000;
67         for(int i=1; i<=row; i++){
68             for(int j=1; j<=col; j++){
69                 scanf("%d",&a[i][j]);
70                 maxx = max(maxx,a[i][j]);
71             }
72         }
73         int ans = fun(); printf("%d\n",ans);
74     }
75 }

```