```
1    /** 2D Maximum Sum
2    ----------------------------------------------------
3    int row,col,a[105][105],cs[105][105],v[105],kd[105];
4    int main(){
5        while(scanf("%d%d",&row,&col)){
6            if(row==0&&col==0)break;
7            for(int i=1; i<=row; i++){
8                for(int j=1; j<=col; j++){
9                    scanf("%d",&a[i][j]);
10               }
11           }
12
13           for(int j=1; j<=col; j++){
14               for(int i=1; i<=row; i++){
15                   cs[j][i]=cs[j][i-1]+a[i][j];
16                   //cs[j][i] means sum of j'th column from (1-to-i) row
17               }
18           }
19
20           int ans=-1000000000;
21           for(int i=1; i<=row; i++){
22               for(int k=i; k<=row; k++){
23                   for(int j=1; j<=col; j++){
24                       v[j] = cs[j][k]-cs[j][i-1];
25                   }
26                   /// kadane or 1D maximum sum
27                   for(int x=1; x<=col; x++) {
28                       kd[x]=kd[x-1]+v[x];
29                       if(kd[x]<0)kd[x]=0;
30                       ans = max(ans,kd[x]);
31                   }
32               }
33           }
34           printf("%d\n",ans);
35       }
36   }
37   /***************************************************************
38   int row,col,a[105][105],rs[105][105],v[105],kd[105];
39   int main(){
40       while(scanf("%d%d",&row,&col)){
41           if(row==0&&col==0)break;
42           for(int i=1; i<=row; i++){
43               for(int j=1; j<=col; j++){
44                   scanf("%d",&a[i][j]);
45               }
46           }
47
48           for(int i=1; i<=row; i++){
49               for(int j=1; j<=col; j++){
50                   rs[i][j]=rs[i][j-1]+a[i][j];
51                   //rs[i][j] means sum of i'th row from (1-to-j) column
52               }
53           }
54
55           int ans=-1000000000;
56           for(int j=1; j<=col; j++){
57               for(int k=j; k<=col; k++){
58                   for(int i=1; i<=row; i++){
59                       v[i] = rs[i][k]-rs[i][j-1];
60                   }
61                   /// kadane or 1D maximum sum
62                   for(int x=1; x<=row; x++){
63                       kd[x]=kd[x-1]+v[x];
64                       if(kd[x]<0)kd[x]=0;
65                       ans = max(ans,kd[x]);
66                   }
67               }
68           }
69           printf("%d\n",ans);
70       }
71   }
```