**DP-S006 (E024)**

Solved Challenges   **0/2**

Back To Challenges List

**Maximum Sum - K*K Sub-Matrix**

**ID:11113     Solved By 637 Users**

The program must accept an integer matrix of size **R*C** and an integer **K** as the input. The program must print the sum of integers in the K*K sub-matrix which has the maximum sum S among the all possible K*K sub-matrices in the given R*C matrix as the output.

**Boundary Condition(s):**
2 <= R, C <= 1000
2 <= K <= R and C

**Input Format:**
The first line contains R and C separated by a space.
The next R lines, each containing C integers separated by a space.
The (R+2)$^{nd}$ line contains K.

**Output Format:**
The first line contains S.

**Example Input/Output 1:**
Input:
4 5
10 20 80 40 55
90 50 90 200 65
60 20 5 20 12
10 50 40 60 8
3

Output:
567

Explanation:
The **3*3** sub-matrix which has the maximum sum is given below.
80 40 55
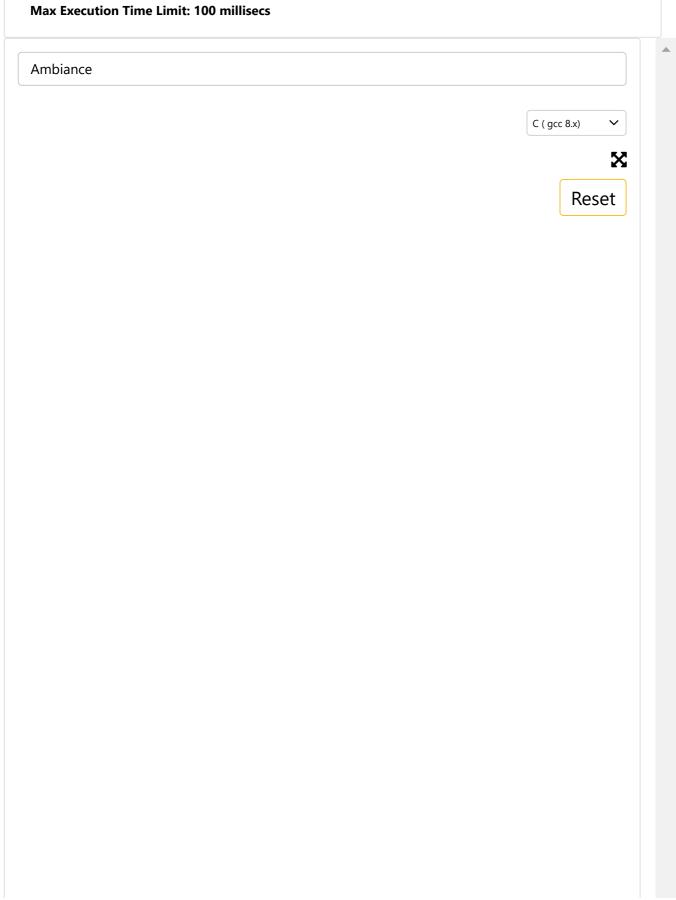90 200 65
5 20 12

**Example Input/Output 2:**
Input:
4 3
4 9 8

2 4 4
5 7 3
7 6 8
2

Output:
25

**Max Execution Time Limit: 100 millisecs**

Ambiance

C ( gcc 8.x)

Reset

```c
1   #include<stdio.h>
2   #include<stdlib.h>
3   #include<limits.h>
4
5   int main()
6   {
7       int R,C;
8       scanf("%d%d", &R,&C);
9       int sum_matrix[R][C+1];
10
11      for(int row=0;row<R;row++)
12          for(int col=0;col<C+1;col++)
13              sum_matrix[row][col] = 0;
14
15      for(int row=0;row<R;row++)
16      {
17          for(int col=1;col<C+1;col++)
18          {
19              int curr;
20              scanf("%d",&curr);
21              sum_matrix[row][col] = curr + sum_matrix[row][col-
22          }
23      }
24
25
26      int k;
27      scanf("%d",&k);
28      int max_sum = INT_MIN;
29      for(int row=0;row<=R-k;row++)
30      {
31          for(int col=1;col<=C-k+1;col++)
32          {
33              int sum = 0;
34              for(int srow=row;srow<row+k; srow++)
35              {
36                  sum+= sum_matrix[srow][col+k-1] - sum_matrix[sr
37              }
38              if(sum>max_sum)
39                  max_sum = sum;
40          }
41      }
42
43      printf("%d", max_sum);
44  }
```

Code did not pass the execution                                    —   ✕

TestCase ID: 63750

Input:

4 5
10 20 80 40 55
90 50 90 200 65
60 20 5 20 12
10 50 40 60 8
3

Expected Output:

567

Your Program Output:

817

Save    Run

☐  Run with a custom test case (Input/Output)