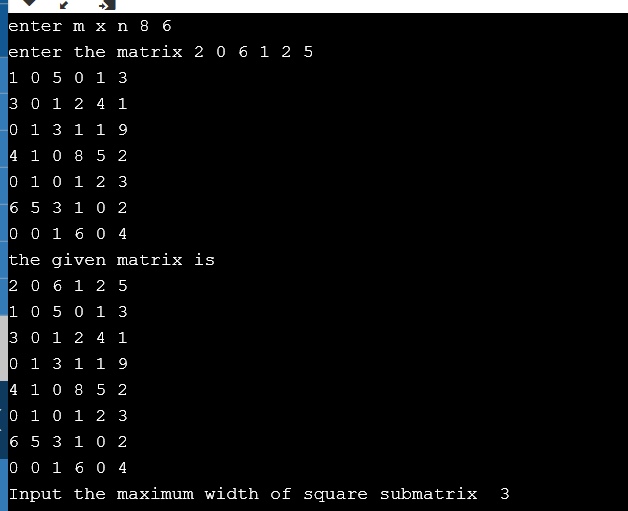
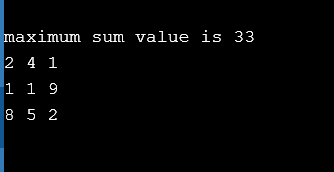
**OUTPUT:**





**ALGORITHM:**

STEP1: Get the input matrix m x n and maximum width of square submatrix (k) from the user.

STEP2: To find all the possible square submatrix for the given m x n matrix.

STEP3: Find the sum of each square submatrix.

STEP4: Find the maximum element in the sum.

STEP5: Note the position of submatrix which has highest sum value and print the corresponding matrix.

**SOURCE CODE(C++)**

#include <iostream>

using namespace std;

int m,n;

int \*pos = NULL;

int main()

{

int grtsum =0, res[3][3], temp[3][3],k,mat[8][6];

cout<< "enter m x n";

cin>>m>>n;

int arr[20];

/\* int mat[8][6] = {

{2,0,6,1,2,5},

{1,0,5,0,1,3},

{3,0,1,2,4,1},

{0,1,3,1,1,9},

{4,1,0,8,5,2},

{0,1,0,1,2,3},

{6,5,3,1,0,2},

{0,0,1,6,0,4}

}; \*/

cout<<"enter the matrix";

for(int i=0;i<m;i++)

for(int j=0;j<n;j++)

cin>>mat[i][j];

cout<<"the given matrix is "<<endl;//printing the given matrix

for(int i=0;i<m;i++){

for(int j=0;j<n;j++)

cout<<mat[i][j]<<" ";

cout<<endl;

}

cout<<"Input the maximum width of square submatrix ";

cin>>k;

for (int i=0; i<m-k+1; i++)

{

// finding square submatrix for given matrix (m x n)

for (int j=0; j<n-k+1; j++)

{

// Calculate and print sum of current sub-square

int sum = 0;

for (int p=i; p<k+i; p++) {

for (int q=j; q<k+j; q++)

{

cout<<mat[p][q]<<" ";

sum = sum+mat[p][q];

if(grtsum<sum){

grtsum = sum;

pos = &(mat[p-(k-1)][q-(k-1)]);

}

}

cout<<endl;

}

cout<<"sum of submatrix is "<<sum;

cout << endl;

}

cout << endl;

}

cout<<"maximum sum value is "<<grtsum<<endl;

for (int i=0; i<k; i++)

{

for (int j=0; j<k; j++)

cout <<\*(pos + i\*n + j) << " ";

cout << endl;

}

return 0;

}