**Maximum sum sub matrix**

#include <iostream>

using namespace std;

#define M 5

#define N 5

typedef pair<int, int> Pt;

Pt maxsum(int mat[][N], int k)

{

int sum[M][N];

sum[0][0] = mat[0][0];

for (int j = 1; j < N; j++)

sum[0][j] = mat[0][j] + sum[0][j - 1];

for (int i = 1; i < M; i++)

sum[i][0] = mat[i][0] + sum[i - 1][0];

for (int i = 1; i < M; i++)

for (int j = 1; j < N; j++)

sum[i][j] = mat[i][j] + sum[i - 1][j] + sum[i][j - 1]

- sum[i - 1][j - 1];

int total, max = INT\_MIN;

Pt p;

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

{

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

{

total = sum[i][j];

if (i - k >= 0)

total = total - sum[i - k][j];

if (j - k >= 0)

total = total - sum[i][j - k];

if (i - k >= 0 && j - k >= 0)

total = total + sum[i - k][j - k];

if (total > max)

max = total, p = make\_pair(i, j);

}

}

return p;

}

int main()

{

int mat[N][N] =

{

{ 3, -4, 6, -5, 1 },

{ 1, -2, 8, -4, -2 },

{ 3, -8, 9, 3, 1 },

{ -7, 3, 4, 2, 7 },

{ -3, 7, -5, 7, -6 }

};

int k = 3;

Pt p = maxsum(mat, k);

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

{

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

cout << mat[i + p.first - k + 1][j + p.second - k + 1] << " ";

cout << '\n';

}

return 0;

}