一个M\*N的矩阵，找到此矩阵的一个子矩阵，并且这个子矩阵的元素的和是最大的，输出这个最大的值。

例如：3\*3的矩阵：

-1 3 -1

2 -1 3

-3 1 2

和最大的子矩阵是：

3 -1

-1 3

1 2

**Input**

第1行：M和N，中间用空格隔开（2 <= M,N <= 500)。   
第2 - N + 1行：矩阵中的元素，每行M个数，中间用空格隔开。(-10^9 <= Mii <= 10^9)

**Output**

输出和的最大值。如果所有数都是负数，就输出0。

**Sample Input**

3 3

-1 3 -1

2 -1 3

-3 1 2

**Sample Output**

7

题解：最大字段和我们可以转化为最大子段和求解。  首先我们枚举行数，在将第i行到第j行的每一列加起来，看成一个数，最后就变成了求（j-i+1）个连续数的序列的最大子段和。 在记录下枚举过程中记录下最大值即可。

#include <iostream>

#include <cstdio>

#include <cstdlib>

#include <cstring>

#include<vector>

#include<algorithm>

using namespace std;

typedef long long ll;

ll a[510][510];

int main()

{

int n,m;

ll x;

cin>>m>>n;

memset(a,0,sizeof(a));

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

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

{

cin>>x;

a[i][j]=a[i-1][j]+x;

}

ll ans(0);

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

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

{

ll sum=0;

for(int k=1;k<=m;k++)

{

sum+=(a[j][k]-a[i-1][k]);

if(sum<0)sum=0;

if(sum>ans)ans=sum;

}

}

cout<<ans<<"\n";

return 0;

}