Task: GRI
Grid

Day 2. Source file `gri.*` *21–04–2008*

Available memory: 32 MB. Maximum running time: 8 s.

The map of Byteland is drawn on a grid of size $n \times m$ (n is the vertical dimension, m is the horizontal dimension). The horizontal lines marking the division are called *parallels*, and are numbered from 0 to n, while the vertical lines of the division are called *meridians*, and are numbered from 0 to m (see figure on the next page).

Weather forecasting is a serious issue in Byteland. For each unit square of the grid a certain amount of computation time is required to prepare the forecast. Due to terrain conditions and other factors this time may vary from square to square. Until very recently the forecasting system was processing the unit squares one after another, so it took as long as the sum of all the unit times to prepare the complete forecast.

You have been asked to design a new system, running on a multiprocessor computer. To share the computations among processors, the area of Byteland should be divided by r parallels and s meridians into $(r+1)(s+1)$ smaller rectangles. Each processor will cover one rectangle of this division and will process the squares of this rectangle one after another. This way the computation time for such rectangle will be the sum of all computation times of the unit squares contained in this rectangle. The computation time of the complete forecast will be the maximum among computation times of the individual processors.

Your task is to find the minimal possible computation time for some choice of r parallels and s meridians.

Task

Write a program, that:

- reads the dimensions of the map of Byteland, the required number of parallels and meridians and the unit computation times from the standard input,
- finds the minimal time required to compute the complete forecast,
- writes the obtained value to the standard output.

Input

The first line of the input contains four integers n, m, r and s, separated by single spaces ($1 \leq r < n \leq 18$, $1 \leq s < m \leq 18$). The following n lines contain the computation times of the unit squares. The j-th number in the $(i+1)$-st line is $c_{i,j}$ — the time required to prepare the weather forecast for the unit square located between the $(i-1)$-st and i-th parallel and between the $(j-1)$-st and j-th meridian ($1 \leq i \leq n$, $1 \leq j \leq m$, $0 \leq c_{i,j} \leq 2\,000\,000$).

Additionally, in test cases worth 40% of points, n and m will not exceed 10.

Output

Your program should write exactly one line. It should contain one integer — the optimal computation time.

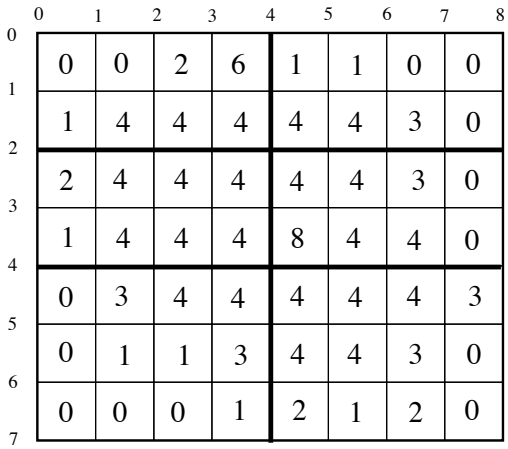
Example

For the input data:

```
7 8 2 1
0 0 2 6 1 1 0 0
1 4 4 4 4 4 3 0
2 4 4 4 4 4 3 0
1 4 4 4 8 4 4 0
0 3 4 4 4 4 4 3
0 1 1 3 4 4 3 0
0 0 0 1 2 1 2 0
```

the correct result is:

```
31
```



The 2-nd and 4-th parallel and the 4-th meridian divide the country into 6 rectangles with computation times 21, 13, 27, 27, 17, 31. The computation time of the complete forecast is 31.