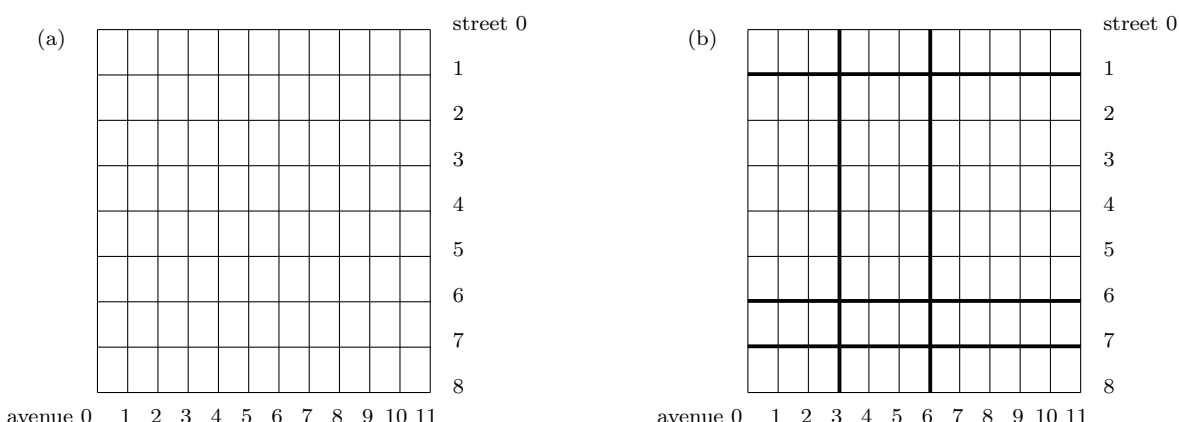


The city *AllSquared* has been designed by many famous architects and engineers. The city lays in a rectangular area of $n \times m$ squared miles where n is the height of the rectangle, and m is its width.

The streets are horizontal divisions distributed uniformly at every one mile, and the avenues are vertical divisions also uniformly distributed at every one mile. Streets are numbered from north to south, starting from zero; avenues from west to east, also starting from zero.

The figure (a) below portrays the city layout for $n = 8$ and $m = 11$. The city is also



divided in horizontal and vertical strips, with the intersection of these strips defining the city subregions (counties). Figure (b) pictures a city divided into 12 counties.

Every county is associated with a vehicle circulation fee, which should be paid every time a car enters the county. If the car origin is a street or avenue delimiting the county, there is no fee. For example, in the above figure a car whose origin is street = 2 and avenue = 3 and whose target is street = 2 and avenue = 9 will have to pay only the county crossing fee relative to avenue 6.

You should write a program that has the following input:

- two strictly positive integers n and m specifying the city dimensions;
- two positive integers h and v specifying the number of horizontal and vertical strips, respectively (for the city in the figure, $h = 4, v = 3$);
- $h - 1$ integers s_1, \dots, s_{h-1} ($1 \leq s_i \leq n - 1$) which denote the streets in which there is a county division line (in the previous figure, the numbers are 1, 6, and 7);
- $v - 1$ integers a_1, \dots, a_{v-1} ($1 \leq a_i \leq m - 1$) which denote the avenues in which there is a county division line (in the the previous figure, the numbers are 3 and 6);
- the circulation fees for each county in the city, i.e., $h * v$ positive integers

$p_{11}, p_{12}, \dots, p_{1v}$
 $p_{21}, p_{22}, \dots, p_{2v}$
 \dots
 $p_{h1}, p_{h2}, \dots, p_{hv}$

where p_{ij} is the circulation fee for the county delimited by a_{i-1} (west), a_{i+1} (east), s_{j-1} (north), and s_{j+1} (south);

- two pairs of positive integers $local_1 = (str_1, av_1)$ and $local_2 = (str_2, av_2)$

The output should be the smallest possible price to pay for going from $local_1$ to $local_2$.

Input

The input format may contain several instances of the problem. Each instance is terminated by a line starting with '%' (percentage symbol).

The description for one instance has the following format:

```
n      m
h      v
s1    s2    ...  sh-1
a1    a2    ...  av-1
p11   p12   ...  p1v
p21   p22   ...  p2v
...
ph1   ph2   ...  phv
w1    t1    w2    t2
%
```

where

- n, m, h, v are integers such that $0 < h \leq n < 100, 0 < v \leq m < 100$;
- s_1, \dots, s_{h-1} are such that $1 \leq s_i \leq n - 1$ for $1 \leq i \leq h - 1$;
- a_1, \dots, a_{v-1} are such that $1 \leq a_i \leq m - 1$ for $1 \leq i \leq v - 1$;
- p_{ij} are positive integers such that $0 < p_{ij} < 10000$;
- w_1, t_1, w_2, t_2 are integers such that $0 \leq w_1, w_2 \leq n$ and $0 \leq t_1, t_2 \leq m$. (w_1, t_1) is the origin and (w_2, t_2) is the target point for the problem;
- there can be an arbitrary number of spaces or empty lines between the numbers in the file.

Output

For an input file with k instances of the problem, the output file should follow the format:

```
c1
c2
...
ck
```

where c_i is the result for the i -th instance in the input file.

Sample Input

```
100 100
3 4
30 60
10 70 80
5 100 1 20
1 100 1 20
1 1 1 20
5 5 10 75
%
8 11
4 3
1 6 7
3 6
10 10 10
10 10 10
10 10 10
10 10 10
2 3 2 9
%
```

Sample Output

```
6
10
```