

D. Big Maximum Sum

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Ahmed and Mostafa used to compete together in many programming contests for several years. Their coach Fegla asked them to solve one challenging problem, of course Ahmed was able to solve it but Mostafa couldn't.

This problem is similar to a standard problem but it has a different format and constraints.

In the standard problem you are given an array of integers, and you have to find one or more consecutive elements in this array where their sum is the maximum possible sum.

But in this problem you are given n small arrays, and you will create one big array from the concatenation of one or more instances of the small arrays (each small array could occur more than once). The big array will be given as an array of indexes (1-based) of the small arrays, and the concatenation should be done in the same order as in this array. Then you should apply the standard problem mentioned above on the resulting big array.

For example let's suppose that the small arrays are $\{1, 6, -2\}$, $\{3, 3\}$ and $\{-5, 1\}$. And the indexes in the big array are $\{2, 3, 1, 3\}$. So the actual values in the big array after formatting it as concatenation of the small arrays will be $\{3, 3, -5, 1, 1, 6, -2, -5, 1\}$. In this example the maximum sum is 9.

Can you help Mostafa solve this problem?

Input

The first line contains two integers n and m , n is the number of the small arrays ($1 \leq n \leq 50$), and m is the number of indexes in the big array ($1 \leq m \leq 250000$). Then follow n lines, the i -th line starts with one integer l which is the size of the i -th array ($1 \leq l \leq 5000$), followed by l integers each one will be greater than or equal -1000 and less than or equal 1000. The last line contains m integers which are the indexes in the big array, and you should concatenate the small arrays in the same order, and each index will be greater than or equal to 1 and less than or equal to n .

The small arrays are numbered from 1 to n in the same order as given in the input. Some of the given small arrays may not be used in big array.

Note, that the array is very big. So if you try to build it straightforwardly, you will probably get time or/and memory limit exceeded.

Output

Print one line containing the maximum sum in the big array after formatting it as described above. You must choose at least one element for the sum, i. e. it cannot be empty.

Please, do not use `%lld` specifier to write 64-bit integers in C++. It is preferred to use `cout` (also you may use `%I64d`).

Examples

input
3 4 3 1 6 -2 2 3 3 2 -5 1 2 3 1 3
output
9

input

```
6 1
4 0 8 -3 -10
8 3 -2 -5 10 8 -9 -5 -4
1 0
1 -3
3 -8 5 6
2 9 6
1
```

output

8