Pokémon Card Convention

CS 4501 Homework
Due: never

You are a Pokémon collector and you have a personal stash of cards. You are attending a virtual Pokémon convention where several vendors are selling their own collections. Each vendor's collection has its own price and you have budgeted at most $\$b$ to spend.

Your goal is to maximize the number of *new* cards you acquire—i.e., cards not already in your own collection. The vendors' collections are all disjoint, meaning no two collections share the same card. But even though the vendors' collections don't overlap with one another, they may overlap with your pre-existing cards. Additionally, the vendors' collections may have multiple instances of the same card, but to you, 1 charmander and 100 charmanders mean the same.

Finally, these collections get very large and your hard drive is running out of space due to all the pokemon games you've illegally emulated. Be sure to use as little memory as possible.

Design an algorithm that chooses which vendor collections to purchase so as to maximize the number of new cards obtained without exceeding your budget.

Hint: Looking at the Knapsack problem and known solutions may be very useful!

Input

- The first line contains two integers b and n: your budget (in dollars) and the number of vendor collections, respectively.
- The next n lines each contain a single integer c_i, the cost of the ith vendor's collection.
- The following line contains the space-separated list of card IDs in *your* existing collection.
- The next n lines each contain the space-separated list of card IDs in the ith vendor's collection.

Output

A single line listing the 1-based indices of the chosen vendor collections, in any order, separated by spaces.

Languages

Your solution *must* be implemented in **Python**, **Java**, or **C++**. Runtime speed and memory usage will be evaluated, and some test cases may involve **millions** of card IDs.

Sample Input

```
30 3
10
15
20
1 3 5
2 4 6 8 10
1 3 7 9
5 6 7 11 12 13
```

Sample Output

```
1 3
```

Explanation. With a budget of \$30, buying collections 1 (cost \$10, new cards $\{2, 4, 6, 8, 10\}$) and 3 (cost \$20, new cards $\{6, 7, 11, 12, 13\}$) yields $5 + 5 = 10$ new cards. No other combination within budget provides more than 10 new cards.

Constraints.

- $1 \leq n \leq 10^5$
- $1 \leq b, c_i \leq 10^9$
- Total number of distinct card IDs across all inputs is at most 10^6.

Your Write-Up. Along with your code, submit a short PDF (1–2 paragraphs) describing your algorithmic approach and data structures. Discuss how you handle checking for "new" cards efficiently and how you optimize the selection under the budget constraint. This is informal and to ensure that you attempted to solve the problem efficiently.