Task: TOY
Toy Ordering Plan

AACPP SuSe 2025 Round 1 Memory: 2MiB 2025.05.06 – 2025.05.13

The birthday of Dexter the Cat is coming up soon and you want to buy **enough toys** to satisfy his excitement. Dexter is pretty strict at judging the quantity of gifts, so you have to carefully select n gifts within the b budget, otherwise either you'll go broke or the fluffy overlord won't be satisfied. There are s dollar-stores to consider, having open hours on d different days. You have limited free time in your schedule, so you want to visit **only one dollar-store**, spend the **least money possible** and you want to go shopping on **only one day**.

For each dollar-store, you know the price p_i of a **single toy** (which, of course, doesn't equal to one dollar) and the number of available toys $a_{i,j}$ for each open hour day out of d. Time is running out, so you better choose the dollar-store and allocate the budget quick! If you don't find any toys within budget, you will have to commit your soul for a full day of unlimited pets, treats, and whatever else Dexter desires.

Input

The first line of input contains four integers n, b, s, d. The s dollar-store descriptions follow, each consisting of two lines:

- first line contains a single integer p_i, the price of the toy at store i,
- second line contains d integers $a_{i,j}$, the number of toys available at store i at day j.

Output

Your program should write a single line containing x, the minimum total cost of the toys or `GIFT MY SOUL` if nothing can be found within budget.

Example

For the input:

```
3 1000 2 3
200
0 2 2
300
27 3 20
```

the correct output is:

```
900
```

while for the input:

```
5 2000 2 4
300
4 3 0 4
450
7 8 0 13
```

the correct output is:

```
GIFT MY SOUL
```

Additional examples

The following initial tests are also available:

- `0b` – sample for Subtask 1, $s = 1$, similar to example above, finding a dollar-store is always possible, $1 \leq n \leq 50$, $1 \leq b \leq 5000$, $1 \leq d \leq 13$;
- `0c` – sample for Subtask 2, $s = 5$, finding a dollar-store is **NOT** always possible, $1 \leq n \leq 100$, $1 \leq b \leq 50000$, $1 \leq d \leq 13$;
- `0d` – sample for Subtask 3, $s = 18$, finding a dollar-store is **NOT** always possible, $1 \leq n \leq 200$, $1 \leq b \leq 500000$, $1 \leq d \leq 13$;

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

In all tests $1 \leq d \leq 13$, $1 \leq p_i \leq 10^4$ and $0 \leq a_{i,j} \leq 10^3$.

Subtask	Limits	Points
1.	$1 \leq n \leq 50$, $1 \leq b \leq 5 \cdot 10^3$, $s = 1$	2
2.	$1 \leq n \leq 100$, $1 \leq b \leq 5 \cdot 10^4$, $1 \leq s \leq 5$	4
3.	$1 \leq n \leq 200$, $1 \leq b \leq 5 \cdot 10^5$, $1 \leq s \leq 18$	4