# Problem Sochi. Reality Show

| | |
|---|---|
| Input file: | `input.txt` or standard input |
| Output file: | `output.txt` or standard output |
| Time limit: | 2 seconds |
| Memory limit: | 512 megabytes |

A popular reality show is recruiting a new cast for the third season! $n$ candidates numbered from 1 to $n$ have been interviewed. The candidate $i$ has aggressiveness level $l_i$, and recruiting this candidate will cost the show $s_i$ roubles.

The show host reviewes applications of all candidates from $i = 1$ to $i = n$ by increasing of their indices, and for each of them she decides whether to recruit this candidate or not. If aggressiveness level of the candidate $i$ is strictly higher than that of any **already accepted** candidates, then the candidate $i$ will definitely be rejected. Otherwise the host may accept or reject this candidate at her own discretion. The host wants to choose the cast so that to maximize the total *profit*.

The show makes revenue as follows. For each aggressiveness level $v$ a corresponding profitability value $c_v$ is specified, which can be positive as well as negative. All recruited participants enter the stage one by one by increasing of their indices. When the participant $i$ enters the stage, events proceed as follows:

- The show makes $c_{l_i}$ roubles, where $l_i$ is initial aggressiveness level of the participant $i$.
- If there are two participants with the same aggressiveness level on stage, they immediately start a fight. The outcome of this is:
  - the defeated participant is hospitalized and leaves the show.
  - aggressiveness level of the victorious participant is increased by one, and the show makes $c_t$ roubles, where $t$ is the new aggressiveness level.
- The fights continue until all participants on stage have distinct aggressiveness levels.

The host wants to recruit the cast so that the total profit is maximized. The profit is calculated as the total revenue from the events on stage, less the total expenses to recruit all accepted participants (that is, their total $s_i$). Help the host to make the show as profitable as possible.

## Input

The first line contains two integers $n$ and $m$ ($1 \le n, m \le 2000$) — the number of candidates and an upper bound for initial aggressiveness levels.

The second line contains $n$ integers $l_i$ ($1 \le l_i \le m$) — initial aggressiveness levels of all candidates.

The thirs line contains $n$ integers $s_i$ ($0 \le s_i \le 5000$) — the costs (in roubles) to recruit each of the candidates.

The fourth line contains $n + m$ integers $c_i$ ($|c_i| \le 5000$) — profitability for each aggrressiveness level.

It is guaranteed that aggressiveness level of any participant can never exceed $n+m$ under given conditions.

## Output

Print a single integer — the largest profit of the show.

## Examples

| input | output |
|---|---|
| 5 4<br>4 3 1 2 1<br>1 2 1 2 1<br>1 2 3 4 5 6 7 8 9 | 6 |
| 2 2<br>1 2<br>0 0<br>2 1 -100 -100 | 2 |
| 5 4<br>4 3 2 1 1<br>0 2 6 7 4<br>12 12 12 6 -3 -5 3 10 -4 | 62 |

## Note

In the first sample case it is optimal to recruit candidates $1, 2, 3, 5$. Then the show will pay $1+2+1+1 = 5$ roubles for recruitment. The events on stage will proceed as follows:

- a participant with aggressiveness level 4 enters the stage, the show makes 4 roubles;
- a participant with aggressiveness level 3 enters the stage, the show makes 3 roubles;
- a participant with aggressiveness level 1 enters the stage, the show makes 1 rouble;
- a participant with aggressiveness level 1 enters the stage, the show makes 1 roubles, a fight starts. One of the participants leaves, the other one increases his aggressiveness level to 2. The show will make extra 2 roubles for this.

Total revenue of the show will be $4 + 3 + 1 + 1 + 2 = 11$ roubles, and the profit is $11 - 5 = 6$ roubles.

In the second sample case it is impossible to recruit both candidates since the second one has higher aggressiveness, thus it is better to recruit the candidate 1.

## Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes **all** tests in this group and all tests of the **required** groups. **Offline evaluation** means that your submission will be evaluated on the tests of the group only after the end of the contest.

| Group | Points | Additional constraints | | Req. groups | Comment |
|---|---|---|---|---|---|
| | | $n$ | $m$ | | |
| 0 | 0 | – | – | – | Example tests. |
| 1 | 14 | $n \leq 15$ | $m \leq 15$ | 0 | |
| 2 | 10 | – | $m = 1$ | – | |
| 3 | 10 | – | $m \leq 2$ | 2 | |
| 4 | 15 | – | $m \leq 5$ | 2, 3 | |
| 5 | 26 | $n \leq 200$ | $m \leq 200$ | 0, 1 | |
| 6 | 25 | – | – | 0, 1, 2, 3, 4, 5 | **Offline evaluation.** |

# Problem Salt Lake City. Present

| | |
|---|---|
| Input file: | `input.txt` or standard input |
| Output file: | `output.txt` or standard output |
| Time limit: | 3 seconds |
| Memory limit: | 512 megabytes |

Catherine received an array of integers as a gift for March 8. Eventually she grew bored with it, and she started calculating various useless characteristics for it. She succeeded to do it for each one she came up with. But when she came up with another one — `xor` of all pairwise sums of elements in the array, she realized that she couldn't compute it for a very large array, thus she asked for your help. Can you do it? Formally, you need to compute

$$\begin{aligned}(a_1+a_2)\oplus(a_1+a_3)\oplus\ldots\oplus(a_1+a_n)\oplus\\ \oplus(a_2+a_3)\oplus\ldots\oplus(a_2+a_n)\oplus\\ \ldots\\ \oplus(a_{n-1}+a_n)\end{aligned}$$

## Input

The first line contains a single integer $n$ ($2 \le n \le 400\,000$) — the number of integers in the array.

The second line contains integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^7$).

## Output

Print a single integer — xor of all pairwise sums of integers in the given array.

## Examples

| input | output |
|---|---|
| 2<br>1 2 | 3 |
| 3<br>1 2 3 | 2 |

## Note

In the first sample case there is only one sum $1+2=3$.

In the second sample case there are three sums: $1+2=3$, $1+3=4$, $2+3=5$. In binary they are represented as $011_2 \oplus 100_2 \oplus 101_2 = 010_2$, thus the answer is 2.

$\oplus$ is the bitwise xor operation. To define $x \oplus y$, consider binary representations of integers $x$ and $y$. We put the $i$-th bit of the result to be 1 when exactly one of the $i$-th bits of $x$ and $y$ is 1. Otherwise, the $i$-th bit of the result is put to be 0. For example, $0101_2 \oplus 0011_2 = 0110_2$.

## Scoring

Tests for this problem are divided into three groups. For each of the groups you earn points only if your solution passes **all** tests in this group and all tests of the **required** groups.

| Group | Points | Additional constraints | | Req. groups | Comment |
|---|---|---|---|---|---|
| | | $n$ | $a_i$ | | |
| 0 | 0 | – | – | – | Example tests. |
| 1 | 34 | $n \le 1000$ | – | 0 | |
| 2 | 37 | – | $1 \le a_i \le 100$ | 0 | |
| 3 | 29 | – | – | 0, 1, 2 | |

# Problem Monaco. Instant Noodles

| | |
|---|---|
| Input file: | `input.txt` or standard input |
| Output file: | `output.txt` or standard output |
| Time limit: | 2 seconds |
| Memory limit: | 512 megabytes |

Wu got hungry after an intense training session, and came to a nearby store to buy his favourite instant noodles. After Wu paid for his purchase, the cashier gave him an interesting task.

> You are given a bipartite graph with positive integers in all vertices of the **right** half. For a subset $S$ of vertices of the **left** half we define $N(S)$ as the set of all vertices of the right half adjacent to at least one vertex in $S$, and $f(S)$ as the sum of all numbers in vertices of $N(S)$. Find the greatest common divisor of $f(S)$ for all possible non-empty subsets $S$.

Wu is too tired after his training to solve this problem. Help him!

## Input

The first line contains a single integer $t$ ($1 \le t \le 500\,000$) — the number of test cases in the given test set. Test case descriptions follow.

The first line of each case description contains two integers $n$ and $m$ ($1 \le n, m \le 500\,000$) — the number of vertices in either half of the graph, and the number of edges respectively.

The second line contains $n$ integers $c_i$ ($1 \le c_i \le 10^{12}$). The $i$-th number describes the integer in the vertex $i$ of the right half of the graph.

Each of the following $m$ lines contains a pair of integers $u_i$ and $v_i$ ($1 \le u_i, v_i \le n$), describing an edge between the vertex $u_i$ of the left half and the vertex $v_i$ of the right half. It is guaranteed that the graph does not contain multiple edges.

Test case descriptions are separated with empty lines. The total value of $n$ across all test cases does not exceed $500\,000$, and the total value of $m$ across all test cases does not exceed $500\,000$ as well.

## Output

For each test case print a single integer — the required greatest common divisor.

## Example

| input | output |
|---|---|
| 3<br>2 4<br>1 1<br>1 1<br>1 2<br>2 1<br>2 2<br><br>3 4<br>1 1 1<br>1 1<br>1 2<br>2 2<br>2 3<br><br>4 7<br>36 31 96 29<br>1 2<br>1 3<br>1 4<br>2 2<br>2 4<br>3 1<br>4 3 | 2<br>1<br>12 |

## Note

The greatest common divisor of a set of integers is the largest integer $g$ such that all elements of the set are divisible by $g$.

In the first sample case vertices of the left half and vertices of the right half are pairwise connected, and $f(S)$ for any non-empty subset is 2, thus the greatest common divisor of these values if also equal to 2.

In the second sample case the subset $\{1\}$ in the left half is connected to vertices $\{1, 2\}$ of the right half, with the sum of numbers equal to 2, and the subset $\{1, 2\}$ in the left half is connected to vertices $\{1, 2, 3\}$ of the right half, with the sum of numbers equal to 3. Thus, $f(\{1\}) = 2$, $f(\{1, 2\}) = 3$, which means that the greatest common divisor of all values of $f(S)$ is 1.

## Scoring

Tests for this problem are divided into three groups. For each of the groups you earn points only if your solution passes **all** tests in this group and all tests of the **previous** groups. **Offline evaluation** means that your submission will be evaluated on the tests of the group only after the end of the contest.

Let $\sum n$ denote the total value of $n$ across all test cases, and $\sum m$ the total value of $m$.

| Group | Points | Additional constraints | | | | Comment |
|---|---|---|---|---|---|---|
| | | $n$ | $m$ | $\sum n$ | $\sum m$ | |
| 0 | 0 | – | – | – | – | Example tests. |
| 1 | 21 | $n \le 20$ | $m \le 400$ | $\sum n \le 100$ | $\sum m \le 2000$ | |
| 2 | 33 | $n \le 5000$ | $m \le 5000$ | $\sum n \le 10\,000$ | $\sum m \le 10\,000$ | |
| 3 | 46 | – | – | – | – | **Offline evaluation.** |

# Problem Rio de Janeiro. Latin Squares

| | |
|---|---|
| Input file: | `input.txt` or standard input |
| Output file: | `output.txt` or standard output |
| Time limit: | 10 seconds |
| Memory limit: | 1024 megabytes |

Chris is a fan of puzzles. Recently he learned about Sudoku puzzles, that are based on Latin squares. A $k \times k$ table is called a *Latin square* if the number of distinct elements in the table is $k$, and there are no two equal elements in the matrix that share the same row or the same column.

For example,

| | |
|---|---|
| A | B |
| B | A |

,

| | | |
|---|---|---|
| S | P | R |
| R | S | P |
| P | R | S |

and

| |
|---|
| Q |

are Latin squares, while

| | | |
|---|---|---|
| A | B | C |
| C | C | A |
| C | A | B |

,

| | | |
|---|---|---|
| A | B | C |
| D | C | A |
| C | A | B |

and

| | | |
|---|---|---|
| A | B | C |
| C | A | B |

are not.

Chris wants to make a new Latin square puzzle. However, he only has an old template, which is an $n \times m$ table. Chris wants to cut a contiguous Latin square fragment from the template. In how many ways can he do this? Two ways to cut a square are considered different if there is a cell that is present in one square, but not present in the other.

## Input

The first line contains two integers $n$ and $m$ — dimensions of the template ($1 \le n, m \le 2\,000$).

The next $n$ lines contain strings $s_i$ that describe the template. Each string $s_i$ contains $2 \cdot m$ characters with ASCII codes between 33 and 126. The cell in row $i$ and column $j$ of the template contains a pair of characters $s_{i,2\cdot j-1}$ and $s_{i,2\cdot j}$ ($1 \le i \le n$, $1 \le j \le m$). Two cells of the template contain equal elements if their ordered character pairs are equal. See the Notes section for further explanation.

## Output

Print a single integer — the number of ways to cut a Latin square from the template.

## Examples

<table>
<tr><th>input</th><th>output</th></tr>
<tr><td>4 5<br>AABBAAAACC<br>BBAABBCCAA<br>AABBCCAABB<br>BBCCAABBCC</td><td>26</td></tr>
<tr><td>5 10<br>!"#$%&'()*+,-./01234<br>56789:;<=>?@ABCDEFGH<br>IJKLMNOPQRSTUVWXYZ[\<br>]^_`abcdefghijklmnop<br>qrstuvwxyz{|}~!"#$%&</td><td>50</td></tr>
</table>

## Note

In the first sample there are 20 ways to cut a $1 \times 1$ Latin square, as well as 6 other ways:

| | | | | |
|---|---|---|---|---|
| AA | BB | AA | AA | CC |
| BB | AA | BB | CC | AA |
| AA | BB | CC | AA | BB |
| BB | CC | AA | BB | CC |

(a) Way 1

| | | | | |
|---|---|---|---|---|
| AA | BB | AA | AA | CC |
| BB | AA | BB | CC | AA |
| AA | BB | CC | AA | BB |
| BB | CC | AA | BB | CC |

(b) Way 2

| | | | | |
|---|---|---|---|---|
| AA | BB | AA | AA | CC |
| BB | AA | BB | CC | AA |
| AA | BB | CC | AA | BB |
| BB | CC | AA | BB | CC |

(c) Way 3

| | | | | |
|---|---|---|---|---|
| AA | BB | AA | AA | CC |
| BB | AA | BB | CC | AA |
| AA | BB | CC | AA | BB |
| BB | CC | AA | BB | CC |

(d) Way 4

| | | | | |
|---|---|---|---|---|
| AA | BB | AA | AA | CC |
| BB | AA | BB | CC | AA |
| AA | BB | CC | AA | BB |
| BB | CC | AA | BB | CC |

(e) Way 5

| | | | | |
|---|---|---|---|---|
| AA | BB | AA | AA | CC |
| BB | AA | BB | CC | AA |
| AA | BB | CC | AA | BB |
| BB | CC | AA | BB | CC |

(f) Way 6

## Scoring

Tests for this problem are divided into five groups. For each of the groups you earn points only if your solution passes **all** tests in this group and all tests of the **required** groups. **Offline evaluation** means that your submission will be evaluated on the tests of the group only after the end of the contest.

| Group | Points | Additional constraints $n, m$ | Req. groups | Comment |
|---|---|---|---|---|
| 0 | 0 | – | – | Example tests. |
| 1 | 9 | $n, m \le 20$ | 0 | |
| 2 | 10 | $n, m \le 100$ | 0, 1 | |
| 3 | 25 | $n, m \le 500$ | 0, 1, 2 | |
| 4 | 26 | – | – | Elements in each row and each column are distinct. |
| 5 | 30 | – | 0, 1, 2, 3, 4 | **Offline evaluation.** |