2025 Canadian Informatics Workshop
Day 1, Problem 1
Course Overload

Time Limit: 3 seconds

Problem Description

Alex is a new student at Consec University, and she needs to choose which courses to take. She has N courses to choose from. The lectures for the i-th course happen daily from day s_i to day f_i (inclusive), beginning at time b_i and ending at time e_i (inclusive), where lecture times are given in minutes from the start of the school day.

Alex must attend all the lectures—if two courses have overlapping lectures, she can't take both of them. Also, some courses are mandatory for graduating and must be taken. Additionally, Alex must take a minimum of M courses to be able to graduate.

Alex assigns a value v_i to each minute of a course to indicate her desire to take that course. The more worthwhile Alex considers one minute of a course to be, the larger the value of v_i she assigns to it. Help Alex maximize the total value of the courses she chooses!

Input Specification

The first line will contain two space-separated integers N and M ($1 \leq N, M \leq 20$).

The next N lines will each contain six space-separated integers s_i, f_i, b_i, e_i, m_i, and v_i ($1 \leq s_i \leq f_i \leq 1\,000$, $1 \leq b_i \leq e_i \leq 1\,000$, $1 \leq v_i \leq 1\,000$, $m_i \in \{0, 1\}$). If $m_i = 0$, the i-th course is optional, and if $m_i = 1$, the i-th course is mandatory.

The following table shows how the available 25 marks are distributed:

Marks Awarded	**Bounds on N**	**Additional Constraints**
2 marks	$1 \leq N \leq 2$	None
3 marks	$1 \leq N \leq 20$	For all i, $m_i = 1$
5 marks	$1 \leq N \leq 20$	For all i, $s_i = f_i$
10 marks	$1 \leq N \leq 15$	None
5 marks	$1 \leq N \leq 20$	None

Output Specification

If Alex cannot graduate because not all mandatory courses can be taken and/or not enough courses in total can be taken, then output -1.

Otherwise, output one integer, the maximum total value of the courses that Alex can fit into her schedule.

Sample Input

```
5 1
1 4 1 50 0 4
1 3 61 150 0 5
1 2 31 60 1 7
2 2 41 50 0 3
1 3 61 150 0 2
```

Output for Sample Input

```
1770
```

Explanation of Output for Sample Input

Since course 3 is mandatory, it must be taken. Course 3 conflicts with courses 1 and 4, so these courses cannot be taken. Courses 2 and 5 do not conflict with course 3, but they do conflict with each other, so only one of courses 2 and 5 can be taken. Courses 2 and 5 have the same total lecture time but course 2 has a greater total value, so it is better to take course 2 than course 5.

By taking the second and third courses, the total value of all courses is $3 \times 90 \times 5 + 2 \times 30 \times 7 = 1\,770$.

2025 Canadian Informatics Workshop

Day 1, Problem 2

Chocolate Bar Partition 2

Time Limit: 5 seconds

Problem Description

Alice and Bob are playing a game with a chocolate bar. Initially, the chocolate bar consists of N connected chocolate squares in a line, where A_i represents the tastiness of the i-th square. Each player takes turns splitting the bar of chocolate, with Alice going first.

For Alice, in a single split operation she can choose any connected piece of chocolate squares and split it into two separate pieces. For Bob, in a single split operation he can choose one of the two connected pieces of chocolate resulting from Alice's split on the previous turn and split his chosen piece into 2 separate pieces. If such a split operation cannot be performed by Bob, his turn is skipped.

Before the start of each player's turn, Alice can choose whether she wants to eat a connected piece of chocolate on the table, ending the game.

Alice's goal is to maximize the sum of tastiness of her chocolate bar, while Bob's goal is to minimize the sum of tastiness that Alice gets. Find the maximum sum of tastiness that Alice can achieve.

Input Specification

The first line of input will consist of a single integer N ($1 \leq N \leq 600$), the number of chocolate squares in the initial bar.

The second line will consist of N space-separated integers $A_1, \ldots, A_N$ ($-10^6 \leq A_i \leq 10^6$), the tastiness of the chocolate squares.

The following table shows how the available 25 marks are distributed:

Marks Awarded	Bounds on N
3 marks	$1 \leq N \leq 8$
7 marks	$1 \leq N \leq 200$
15 marks	$1 \leq N \leq 600$

Output Specification

Output a single integer, the maximum sum of tastiness that Alice can achieve.

Sample Input

```
4
1 2 -4 4
```

Output for Sample Input

```
4
```

Explanation of Output for Sample Input

Alice will split the bar into $[1, 2, -4]$ and $[4]$, then eat the piece with tastiness 4. It can be shown that 4 is the maximum tastiness she can achieve.

2025 Canadian Computing Olympiad
Day 1, Problem 1
2025 Canadian Informatics Workshop
Day 1, Problem 3

Asteroid Mining

Time Limit: 3 seconds

Problem Description

It is the year 2217 and Ryan is an asteroid miner. He makes a living by mining asteroids and selling them at the CCO (Celestial Cargo Outpost).

On his latest mining expedition, he has mined N mineral chunks where the i-th chunk has a value v_i and a mass m_i. Ryan plans to transport a set of chunks to the CCO with his rocket, but he only has enough fuel to last one more trip. He calculated that the maximum total mass he can safely carry on his rocket is M. Due to Ryan's mining technique, the chunks exhibit a special property: for any two mineral chunks, one's mass is divisible by the other chunk's mass.

Help Ryan find the maximum total value he can ship to CCO while adhering to his rocket's constraints.

Input Specification

The first line will contain two space-separated integers N ($1 \leq N \leq 500\,000$) and M ($1 \leq M \leq 10^{12}$).

The next N lines will each contain two space-separated integers v_i ($1 \leq v_i \leq 10^{12}$) and m_i ($1 \leq m_i \leq 10^{12}$), representing the value and mass of the i-th mineral chunk respectively. **Additionally, for any two mineral chunks i, j ($1 \leq i, j \leq N$), either $m_i \mid m_j$ or $m_j \mid m_i$, where $a \mid b$ means that a is a divisor of b (i.e., b/a is an integer).**

The following table shows how the available 25 marks are distributed:

Marks Awarded	Bounds on N	Bounds on M	Additional Constraints
2 marks	$N = 2$	$1 \leq M \leq 10^4$	None
2 marks	$1 \leq N \leq 20$	$1 \leq M \leq 10^4$	None
4 marks	$1 \leq N \leq 1\,000$	$1 \leq M \leq 10^4$	None
6 marks	$1 \leq N \leq 1\,000$	$1 \leq M \leq 10^8$	None
2 marks	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10^8$	All m_i are equal.
3 marks	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10^8$	At most 2 distinct m_i.
6 marks	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10^{12}$	None

Output Specification
On one line, output one integer, the maximum total value Ryan can ship to CCO.

Sample Input
```
6 10
1 1
5 2
200 6
9 2
6 2
100 1
```

Output for Sample Input
```
310
```

Explanation of Output for Sample Input
Ryan can take all the chucks except the second and fifth chucks to achieve a total value of $1 + 200 + 9 + 100 = 310$. Note that the total mass of the chunks is $1 + 6 + 2 + 1 = 10$. We can show that this is optimal.

2025 Canadian Informatics Workshop
Day 2, Problem 1
Hit the Griddy

Time Limit: 2 seconds

Problem Description
Ernest has a special dartboard that can be represented as an $N \times M$ grid, where N and M are odd. The rows are numbered from 1 to N, and the columns from 1 to M. For the cell in the i^{th} row and j^{th} column, we denote its value by $v_{i,j}$. For this specific grid, $v_{i,j}$ equals the concatenation of i and j. Some examples are $v_{12,434} = 12434$ and $v_{90,10} = 9010$.

Ernest wants to throw a dart to hit the center of the dartboard. However, the center of this dartboard is defined a bit strangely; instead of the geometric center, it is defined as the cell containing the median value in the grid.

Help Ernest determine what the median value is!

Input Specification
The first and only line of input contains two space-separated integers N and M. It is guaranteed that N and M are odd.

The following table shows how the available 25 marks are distributed:

Marks Awarded	**Bounds on N**	**Bounds on M**
2 marks	$N = 1$	$1 \leq M \leq 10^9$
3 marks	$1 \leq N \leq 10$	$1 \leq M \leq 10$
4 marks	$1 \leq N \leq 2\,000$	$1 \leq M \leq 2\,000$
8 marks	$1 \leq N \leq 100\,000$	$1 \leq M \leq 10^9$
8 marks	$1 \leq N \leq 10^9$	$1 \leq M \leq 10^9$

Output Specification
On a single line, output the median value of all values in the grid.

Sample Input 1
```
3 5
```

Output for Sample Input 1
```
23
```

Explanation of Output for Sample Input 1
The dartboard looks like:

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35

The median value is 23.

Sample Input 2

```
1023 957
```

Output for Sample Input 2

```
453512
```

2025 Canadian Informatics Workshop
Day 2, Problem 2
Polling Stations

Time Limit: 2 seconds

Problem Description

Election time is coming up in Kitchener! After having retired from her mayoral duties, this year Alanna is serving as the Chief Electoral Officer for the city. Things have been going well, but there is still one major issue—the polling stations have no software!

Thus, she's had to quickly assemble a team of programmers to program the polling stations before election day. You've been selected to be one of the programmers on the team, and have been tasked with writing a part of the polling software. Your goal is to write a program to multiply 2 numbers.

However, don't be fooled by the simplicity of this task! The polling stations are ancient, and so are only programmable in an archaic, obscure language called APL (Advanced Polling Language). APL only supports binary variables, along with the 4 binary operations AND, OR, NOT, and XOR (which can be applied on said variables). Are you up for the challenge?

Formally, you are given the constraints N and M, and your goal is to output an APL program that inputs an N-bit binary number x and an M-bit binary number y, and outputs their product $x \times y$ as an $(N+M)$-bit binary number. Moreover, as the polling station computers are very slow, your program may also only contain up to C instructions in total.

For the full output specification, please see the Output Specification section below.

Note on grading: The APL program that your code outputs will be graded on multiple test cases. Each test case in turn will simulate its execution on multiple pairs (x, y) of inputs. If the APL program does not compute the correct result on any input pair, it will be considered incorrect for that test case. Unfortunately, the electoral committee does not have the resources to test your program on a real polling station.

Input Specification

The first and only line of input contains 3 integers, N, M, and C.

The following table shows how the available 25 marks are distributed:

Marks Awarded	Bounds on N	Bounds on M	Bounds on C
2 marks	$N = 2$	$M = 1$	$C = 2 \cdot 10^6$
4 marks	$N \leq 200$	$M \leq 2$	$C = 2 \cdot 10^6$
4 marks	$N \leq 200$	$M \leq 6$	$C = 2 \cdot 10^6$
9 marks	$N \leq 200$	$M \leq 200$	$C = 2 \cdot 10^6$
3 marks	$N \leq 200$	$M \leq 200$	$C = 500\,000$
3 marks	$N \leq 200$	$M \leq 200$	$C = 300\,000$

Output Specification
Output a valid APL program. The language specification for APL is as follows:

On the first line, output an integer c ($1 \leq c \leq C$), the number of instructions in the program.

The next c lines should each contain one gate. Each gate should be a line in one of the following 4 forms:

```
AND <in1> <in2>
OR <in1> <in2>
XOR <in1> <in2>
NOT <in1>
```

The parameters `<in1>` and `<in2>` are variable identifiers that correspond to the input of each gate. APL supports only binary variables, each of which are identified by an integer.

Moreover, note that the instructions in your program will be executed in a sequential order. This means that the parameters `<in1>` and `<in2>` must use existing variables declared as the output in previous instructions. How variables are defined/interpreted is described below.

The first N variables (variables 0 to $N-1$) correspond to the bits of the first input number x from the least to most significant bit, and the next M variables (variables N to $N+M-1$) correspond to the bits of the second input number y in the same order.

Next, the output of every gate will each create a new variable starting from the variable $N + M$ (e.g. the first gate outputs to the variable $N + M$, the second gate outputs to $N + M + 1$, etc.).

Finally, the last $N + M$ variables (which are the outputs of the last $N + M$ gates) will be interpreted as the binary representation of the product z (where $z = x \times y$) in order from least significant bit to most significant bit. Consequently, if your program has less than $N + M$ gates, then it will be judged as incorrect.

Sample Input
```
1 1 2000000
```

Output for Sample Input

```
5
OR 0 1
NOT 2
AND 3 3
NOT 4
XOR 4 4
```

Explanation of Output for Sample Input

The outputted APL program computes the bitwise OR of the input numbers x and y and stores it in the output number z. Note that this program incorrectly computes $x \times y$ and is for demonstration purposes only.

Additionally, note that since x and y are both 1-bit numbers, the output number z is expected to be $1 + 1 = 2$ bits long, including any leading zeroes.

2025 Canadian Computing Olympiad
Day 2, Problem 1
2025 Canadian Informatics Workshop
Day 2, Problem 3

Restaurant Recommendation Rescue

Time Limit: 2 seconds

Problem Description

A certain aspiring musician K loves going for shabu-shabu! Recently, she's been to N shabu-shabu restaurants, numbered $1, 2, \ldots, N$, following the following algorithm:

1. K keeps an ordered list of recommendations, starting with restaurant 1.
2. On the i-th day, she visits the next recommended restaurant on her list, which recommends her restaurants $R_i = \{r_{i,1}, \ldots, r_{i,\ell_i}\}$.
3. K appends R_i to her list of restaurants to visit.
4. K repeats steps 2-4 until she runs out of recommended restaurants.
5. K writes down the array $A_0, \ldots, A_{N-1}$, where A_i equals the number of restaurants she was recommended on the $(i+1)$-th day. That is, $A_i = |R_{i+1}|$.

It is guaranteed that $\bigcup_{i=1}^{N} R_i = \{2, \ldots, N\}$ and $R_i \cap R_j = \varnothing$ for $i \neq j$, that is, every restaurant, other than the first, will be recommended by exactly one other restaurant.

Once K finishes her list, K's delinquent friend H decides to play a prank on her! She replaces the array $A_0, \ldots, A_{N-1}$ with another array $B_0, \ldots, B_{N-1}$! K thinks that this new array B_i might just be a cyclic shift of her array, so she asks you to determine all possible $0 \leq k < N$ such that $A_i = B_{(i+k) \bmod N}$, for all $0 \leq i < N$ and **any** valid output of her algorithm $A_0, \ldots, A_{N-1}$.

Furthermore, K will then perform Q operations, where for the i-th operation, she swaps B_{x_i}, B_{y_i} and asks you to do the same on the new array. Can you help K see through her friend's prank?

Input Specification

The first line of input will contain two integers, N ($1 \leq N \leq 500\,000$) and Q ($0 \leq Q \leq 300\,000$).

The next line of input will contain N space-separated non-negative integers, $B_0, B_1, \ldots, B_{N-1}$ ($0 \leq B_i < N$), the initial sequence.

The i-th of the next Q lines of input will contain two integers each, x_i and y_i ($0 \leq x_i, y_i < N$ and $x_i \neq y_i$), indicating you are to swap B_{x_i} with B_{y_i}.

The following table shows how the available 25 marks are distributed:

Marks Awarded	**Bounds on N**	**Bounds on Q**
3 marks	$1 \leq N \leq 8$	$Q = 0$
7 marks	$1 \leq N \leq 5\,000$	$Q = 0$
10 marks	$1 \leq N \leq 500\,000$	$Q = 0$
5 marks	$1 \leq N \leq 500\,000$	$0 \leq Q \leq 300\,000$

Output Specification

For each of the $Q + 1$ arrays (including the initial array $B_0, \ldots, B_{N-1}$), let $S = \{k_1, \ldots, k_m\}$ denote the set of integers $0 \leq k_j < N$ such that there exists a valid output $A_0, \ldots, A_{N-1}$ of K's algorithm such that $A_i = B_{(i+k_j) \bmod N}$ for all $0 \leq i < N$. Output, on a single line, the integers m and $\sum_{i=1}^{m} k_i \pmod{998\,244\,353}$, separated by a space.

In particular, if $S = \varnothing$, your output should be `0 0`.

Sample Input

```
5 3
1 2 0 0 1
0 2
1 3
3 2
```

Output for Sample Input

```
1 4
1 1
1 2
1 2
```

Explanation of Output for Sample Input

The array A is $[1, 1, 2, 0, 0]$; it can be shown this is the only valid output of K's algorithm that corresponds to the array $B = [1, 2, 0, 0, 1]$. One input for K's algorithm that yields this array A is:

$$R_1 = \{2\}$$
$$R_2 = \{3\}$$
$$R_3 = \{4, 5\}$$
$$R_4 = \varnothing$$
$$R_5 = \varnothing.$$

After swapping B_0 and B_2, we get the array

$$B = [0, 2, 1, 0, 1].$$

It can be shown the only valid output of K's algorithm that corresponds to this is

$$A = [2, 1, 0, 1, 0].$$

One possible input to K's algorithm that yields this array A is

$$\begin{aligned} R_1 &= \{2, 3\} \\ R_2 &= \{4\} \\ R_3 &= \varnothing \\ R_4 &= \{5\} \\ R_5 &= \varnothing. \end{aligned}$$

Tips for Python (CIW Only) You are recommended to use fast input (for example, `sys.stdin.read()` and `sys.stdout.write()`) if you are attempting the final subtask.