

Wesley's Anger Contest Reject 3 - Christmas Carols

Time limit: 4.0s **Memory limit:** 256M

After hearing your friends sing some Christmas songs, you decided to prepare for next Christmas by adding some songs to your music playlist. Your playlist begins with N songs. Each song has an integer rating r that you've assigned to it and is composed of a series of k notes, which are represented by a sequence of integers $a_1, a_2, \dots, a_k$.

Over time, Q events of two different types will occur:

- A new song is added to your playlist. This song will also have a rating of r and is a series of k notes $(a_1, a_2, \dots, a_k)$.
- One of your friends sings k notes $(a_1, a_2, \dots, a_k)$. When this happens, you want to know if the song they are singing could match a song in your playlist. Since your friend may sing off-key, you may need to consider the transposition of their song. A transposition of a song is when the same integer is added to each note in the song. Your friend's song matches a song in the playlist if the notes in any transposition of your friend's song are a contiguous subsequence of the notes in the playlist's song. Formally, let the notes in your friend's song be $a_1, a_2, \dots, a_k$ and the notes in the playlist's song be $b_1, b_2, \dots, b_l$. Your friend's song matches the playlist's song if there exists an integer m where $1 \leq m \leq l - k$ and an integer c such that $a_i + c = b_{m+i}$ holds for all $1 \leq i \leq k$. You want to know the maximum rating out of all songs that match your friend's song.

Constraints

For this problem, you will NOT be required to pass all the samples in order to receive points. In addition, all subtasks are disjoint, and you are NOT required to pass previous subtasks to earn points for a specific subtask.

For all subtasks:

$$N, Q \geq 0$$

$$1 \leq r \leq 1\,000\,000$$

$$1 \leq k \leq 500\,000$$

$$1 \leq a_i \leq 1\,000\,000 \text{ for all } 1 \leq i \leq k$$

The sum of the number of notes in the songs in your playlist after all Q events will not exceed 500 000.

The sum of the number of notes sung by your friends after all Q events will not exceed 500 000.

Subtask 1 [6%]

The sum of the number of notes in the songs in your playlist after all Q events will not exceed 500.

The sum of the number of notes sung by your friends after all Q events will not exceed 500.

Subtask 2 [46%]

There will be no events of the first type.

Subtask 3 [48%]

No additional constraints.

Input Specification

The first line will contain two integers, N and Q , representing the number of songs initially in your playlist and the number of events.

The next N lines describe the initial songs. Each line begins with two integers, r and k , representing the rating and the number of notes in this song. k integers follow $(a_1, a_2, \dots, a_k)$ representing the notes in the song.

Some of the input will be given in encrypted format. Instead of each variable x being given, a variable x' will be given instead. To decrypt the input, you should take the bitwise xor of the last value that was outputted, and x' , that is $x = x' \oplus lastAns$. If no previous value was outputted, then $lastAns = 0$.

The next Q lines describe the event. Each line begins with a single integer that is either `1` or `2`, representing the type of event.

- For events of the first type, the next two integers are r and k , representing the rating and the number of notes in this song to be added to your playlist. k integers follow $(a'_1, a'_2, \dots, a'_k)$ representing the notes in the song. Note that the values $a'_1, a'_2, \dots, a'_k$ are given in encrypted format.
- For events of the second type, the next integer is k , representing the number of notes sung by your friend. k integers follow $(a'_1, a'_2, \dots, a'_k)$ representing the notes sung by your friend. Note that the values $a'_1, a'_2, \dots, a'_k$ are given in encrypted format.

Output Specification

This problem is graded with an `identical` checker. This includes whitespace characters. Ensure that every line of output is terminated with a `\n` character and that there are no trailing spaces.

After each event of the second type, output the maximum rating out of all songs that match your friend's song. If there are no songs that match, output `0` instead.

Sample Input 1

```
1 3
3 4 1 2 3 4
2 3 5 6 7
1 5 6 8 15 14 13 12 19
2 3 6 5 4
```

Sample Input 1 Decrypted

For your convenience, below is a version of Sample Input 1 that is NOT encrypted. Note that all of the actual test files will be encrypted in the format specified above.

```
1 3
3 4 1 2 3 4
2 3 5 6 7
1 5 6 11 12 13 14 15 16
2 3 5 6 7
```

Sample Output 1

```
3
5
```

Sample Input 2

```
2 3
4 1 100
2 3 10 1 20
2 1 50
2 2 86 97
2 3 8 22 28
```

Sample Input 2 Decrypted

For your convenience, below is a version of Sample Input 2 that is NOT encrypted. Note that all of the actual test files will be encrypted in the format specified above.

```
2 3
4 1 100
2 3 10 1 20
2 1 50
2 2 82 101
2 3 10 20 30
```

Sample Output 2

```
4
2
0
```