

1 Introduction

In this project, the task is to create the optimal playlist by comparing the song selections of people in a group based on their inputs.

2 Details

You have gathered N playlists (provided as input). Each playlist has its own set of unique songs. Moreover, no two playlists share the same songs; each one is distinct.

Every song within each playlist is categorized into three vibe-worthy categories: *Heartache*, *Roadtrip*, and *Blissful*. A song is assigned three scores based on its relevance to the listed categories, ranging between $[0, 100]$.

Each song has a play count, which remains unchanged throughout crafting EpicBlend.

The ultimate playlist you are creating from your friends' playlists, EpicBlend, should include at most a specified number of songs from each category, denoted as c_h , c_r , and c_b .

EpicBlend should include songs with the highest category scores among the songs in the playlists you collected from your friends' accounts. In the event of a tie, songs with lexicographically smaller names should be prioritized. Remember, **EpicBlend cannot contain duplicate songs**; each song must be distinct.

However, a song **might be included under two or three different categories** in the playlist at the same time **if it achieves a sufficient score in both**.

Additionally, it is important to bear in mind that there is a limit, denoted as l , for the number of songs that can be added to each category in every playlist. For example, if the limit is set at 10, each playlist can individually add up to 10 songs in each category.

Once again, if the same song can enter EpicBlend under two categories, it will **concurrently decrease** the remaining capacity for **both** of these categories in the playlist.

Here is what might change in your friends' playlists:

1. A new song j can be added to playlist i .
2. A song j can be removed from its playlist.

These events may cause a change in the EpicBlend. Some new songs will enter and some songs will leave the EpicBlend, after the changes in your friends' playlists. So, your structure

must handle these events in an efficient way.

Finally, you provided your friends with five chances to request information about EpicBlend. When your friends ask, you'll share all EpicBlend songs organized by their play count in descending order. In case of tie, songs will be listed lexicographically by their names.

3 Input & Output

3.1 Input

- There will be 2 input files named `<song>` and `<test-case>`. A single song file will be utilized for more than one test case.

3.1.1 Song File

- This file contains a database of songs you will use for various cases.
- The first line contains the number of songs, denoted as S .
- The following S lines contain songs, presenting their song ID, play count, and scores for the three categories, all organized in sequential order.
- The structure of the songs is as follows:

`<song-id><song-name><play-count><heartache-score><roadtrip-score><blissful-score>`

Example: 1 Song1 100 100 0 50

Explanation: Song1 has an ID of 1, with a play count 100. Its Heartache score is 100, Roadtrip score is 0, and Blissful score is 50.

- It is **not mandatory** for all songs in the song file to be present in a playlist. There can be songs left out.

Song File					
5					
1	song1	100	99	0	50
2	song2	200	80	10	20
3	song3	50	70	20	30
4	song4	30	10	60	70
5	song5	10	50	50	50

Table 1: Sample Song File

3.1.2 Test Case File

- This file contains the case that you are working on.
- The first line includes four integers: l , indicating the category addition limit for each playlist, and c_h , c_r , and c_b , representing the song counts for the EpicBlend you are crafting based on each of the three categories.

<playlist-category-limit> <heartache-category-limit><roadtrip-category-limit><blissful-category-limit>

Example: 4 3 8 10

Explanation: Each playlist can contribute EpicBlend a maximum of 4 songs to each category, meaning you can add up to 4 Heartache, 4 Roadtrip, and 4 Blissful songs from each playlist. Moreover, EpicBlend can contain a maximum of 3 Heartache songs, 8 Roadtrip songs, and 10 Blissful songs.

- The second line contains the number of playlists, denoted as N .
- The subsequent $2N$ lines consist of pairs. Each pair starts with two integers: the playlist ID and the number of songs in that playlist. The second line contains the song IDs represented by integers, and their count is the number of songs in that playlist. The properties of these songs can be obtained from the song database file.

<playlist-id><playlist-song-count>
<song-1><song-2>...<song-playlist-song-count>

Example: 2 5
4 6 78 65 100

Explanation: The playlist with ID 2 has 5 songs at the beginning, consisting of songs with IDs 4, 6, 78, 65, and 100.

- After this part, the operations come.
- The following line contains the number of events, denoted as E .
- The next E lines contain one of the three event types: ADD, REM, or ASK.

ADDITION EVENT:

ADD <song-id> <playlist-id>

One of your friends adds the song with ID <song-id> to Playlist ID <playlist-id>.

You should add the chosen song to the playlist, which might change EpicBlend, too.

For instance, if the added song's Heartache score is higher than the score of the previously added song in EpicBlend's Heartache category, the existing song in EpicBlend should be replaced with the newly added song in the playlist.

Note: It is guaranteed that the song is not currently on any playlists.

REMOVAL EVENT:

REM <song-id> <playlist-id>

One of your friends removes the song with ID <song-id> from Playlist ID <playlist-id>.

You should remove the selected song from the playlist, which might also impact EpicBlend. For example, if the removed song is already in EpicBlend in a category, it should be replaced with the next highest-scored song from the entire set of playlists that have not reached the limit *l*. Notice that it is possible that the removed song might be replaced by a song that is counted for another category in EpicBlend.

Note: The song is guaranteed to be in the given playlist.

ASKING EVENT:

ASK

One of your friends asks you to print the EpicBlend.

You should print EpicBlend in descending order of play counts. In the case of tied counts, the song with lexicographically smaller names is printed first.

Note: It is given that the ASK event will not occur more than 5 times.

Test Case File	Explanation
1 2 2 2	Each playlist you have gathered can contribute at most 1 song to Epic Blend in each category. EpicBlend can include up to 2 Heartache songs, 2 Roadtrip songs, and 2 Blissful songs in total.
2	There are 2 playlists.
1 2	Playlist with ID 1 contains 2 songs.
1 2	Playlist with ID 1 contains songs with IDs 1 and 2.
2 3	Playlist with ID 2 contains 3 songs.
3 4 5	Playlist with ID 2 contains songs with IDs 3, 4 and, 5.
3	There will be 3 events.
REM 3 2	One of your friends removes the song with ID 3 from Playlist ID 2.
ADD 3 1	One of your friends adds the song with ID 3 to Playlist ID 1.
ASK	One of your friends asks you to print the EpicBlend.

Table 2: Sample Test Case File with Explanations

3.2 Output

- At the beginning, you must create EpicBlend using the playlists you’ve collected from your friends. During this step, there is no need to print anything.
- After each ADD or REMOVE event, print the changes in EpicBlend across three categories. Output 2 lines, each containing 3 integers. The first line should represent the ID of added songs for each category, and the second line should represent the ID of removed songs for each category. Remember that the addition and removal in EpicBlend are based on score differences, and EpicBlend always includes the highest-scored possible songs in each category.

```
<added-heartache><added-roadtrip><added-blissful>
<removed-heartache><removed-roadtrip><removed-blissful>
```

Example: 0 0 1
0 0 2

Explanation: The song with ID 2 is removed from Blissful category of EpicBlend because a higher-scored song with ID 1 now exists in the Blissful category. Consequently, the song with ID 1 is added to EpicBlend. No changes occurred in the

Heartache and Roadtrip categories.

- User input **does not involve** the addition or removal of songs directly to the EpicBlend. **Do not forget, EpicBlend changes are the events' consequences.**
- After each ASK event, you should print the song IDs of the songs in EpicBlend, following the given rules for order, in a single line with IDs separated by spaces.

So the examination and final output of the given input is:

At the beginning, there are two playlists with IDs 1 and 2.

Playlist 1	Playlist 2
1 song1 100 99 0 50	3 song3 50 70 20 30
2 song2 200 80 10 20	4 song4 30 10 60 70
	5 song5 10 50 50 50

	Heartache	Roadtrip	Blissful
Added to EpicBlend	song1(99) song3(70)	song4(60) song2(10)	song4(70) song1(50)
Added From Playlist 1:	1	1	1
Added From Playlist 2:	1	1	1
Not Added Songs	song2(80) song5(50) song4(10)	song5(50) song3(20) song1(0)	song5(50) song3(30) song2(20)

Firstly, let us examine the Heartache category. We start with the highest-scored song, Song 1 from Playlist 1. We include Song 1 in the EpicBlend. EpicBlend allows up to 2 Heartache songs, so we check the second-highest scoring song, Song 2. However, there's a limit on playlists: in this case, only 1 song from each playlist can be added under each category. Since Song 2 is in the same playlist as Song 1, we skipped it and chose the third-ranking song, Song 3. Then, we move on to the next category.

Now we focus on the Roadtrip category, beginning with the top-scoring song, Song 4 from Playlist 2. We include Song 4 in the EpicBlend. EpicBlend permits the inclusion of up to 2 songs in the Roadtrip category. Moving on to assess the second-highest scoring song, namely Song 5, we encounter a limitation on playlists—only 1 song from each category can be added per playlist. As Song 5 shares a playlist with Song 4, we bypass it and move to the third-ranking song, Song 3. However, Song 3, too, shares a playlist with Song 4. So, we choose the fourth-best song, Song 2, because it is from a different playlist, Playlist 1. Before adding Song 2 to the Roadtrip category, we ensure that Playlist 1 has not surpassed its limit for this category, and upon confirmation, we

proceed with the inclusion of Song 2.

Lastly, we explore the Blissful category. Beginning with the top-scoring song, Song 4 from Playlist 2, we add it to EpicBlend. Following that, we consider the second-highest scoring song, Song 1 from Playlist 1. After confirming that Playlist 1 and Playlist 2 haven't exceeded their category limits, we include these songs in EpicBlend.

An important note: observe that Song 4 and Song 1 are included in EpicBlend under two categories, yet they are listed only once. EpicBlend avoids duplicate songs but permits songs to be included from multiple categories.

The songs in EpicBlend are arranged based on their play count, with lexicographic order used as a tiebreaker for songs with the same play count.

Taking these into account, EpicBlend looks like this:

EpicBlend					
2	song2	200	80	10	20
1	song1	100	99	0	50
3	song3	50	70	20	30
4	song4	30	10	60	70

Event 1: REM 3 2

Playlist 1	Playlist 2
1 song1 100 99 0 50	3 song3 50 70 20 30
2 song2 200 80 10 20	4 song4 30 10 60 70
	5 song5 10 50 50 50

	Heartache	Roadtrip	Blissful
Added to EpicBlend	song1(99) song3(70) song5(50)	song4(60) song2(10)	song4(70) song1(50)
Added From Playlist 1:	1	1	1
Added From Playlist 2:	1	1	1
Not Added Songs	song2(80) song4(10)	song5(50) song3(20) song1(0)	song5(50) song3(30) song2(20)

In this event, Song 3 is removed from Playlist 2. In the Heartache category, Song 3 is removed. So, we must replace it with the highest-scoring suitable song, Song 5. We

ensure that neither playlist exceeds its limit. The other categories are not affected by this event.

Output for REM 3 2
5 0 0
3 0 0

As can be seen in the previous table, Song 5 entered the Heartache category, and Song 3 was removed from the Heartache category. And there is no change in other categories.

EpicBlend
2 song2 200 80 10 20
1 song1 100 99 0 50
3 song3 50 70 20 30
4 song4 30 10 60 70
5 song5 10 50 50 50

Event 2: ADD 3 1

Playlist 1	Playlist 2
1 song1 100 99 0 50	4 song4 30 10 60 70
2 song2 200 80 10 20	5 song5 10 50 50 50
3 song3 50 70 20 30	

	Heartache	Roadtrip	Blissful
Added to EpicBlend	song1(99) song5(50)	song4(60) song3(20) song2(10)	song4(70) song1(50)
Added From Playlist 1:	1	1	1
Added From Playlist 2:	1	1	1
Not Added Songs	song2(80) song3(70) song4(10)	song5(50) song2(10) song1(0)	song5(50) song3(30) song2(20)

In this event, Song 3 is added to Playlist 1. Let us examine Roadtrip category. Song 3 has a higher score than Song 2. Thus, we should replace Song 2 with Song 3. We can do this operation because both are now on the same list, and no playlist exceeds their limit. In the Heartache category, Song 3 cannot replace Song 5 because Song 1 scores better in the same playlist.

Output for ADD 3 1
0 3 0
0 2 0

Song 3 entered the Roadtrip category and Song 2 was removed from the Roadtrip category. And there is no change in other categories.

EpicBlend
2 song2 200 80 10 20
1 song1 100 99 0 50
3 song3 50 70 20 30
4 song4 30 10 60 70
5 song5 10 50 50 50

Event 3: ASK

Output for ASK
1 3 4 5

Output is listed from the most played song to the least played song.

Output File
5 0 0
3 0 0
0 3 0
0 2 0
1 3 4 5

Table 3: Sample 1 Output File

3.3 More Examples

Song File	Test Case File
12	2 1 3 6
1 song1 100 99 0 50	3
2 song2 200 80 10 20	1 2
3 song3 50 70 20 30	3 5
4 song4 30 10 60 70	2 1
5 song5 10 50 50 50	4
6 song6 5 10 70 70	3 1
7 song7 44 90 0 100	1
8 song8 35 0 100 0	10
9 song9 4 0 20 60	ADD 9 2
10 song10 100 80 0 90	ASK
11 song11 300 0 80 80	ADD 10 2
12 song12 10000 100 100 100	ASK
	ADD 11 2
	ASK
	ADD 12 2
	ASK
	REM 5 1
	ASK

Table 4: Sample 2 Input Files

Output File
0 0 9
0 0 0
1 3 4 5 9
0 0 10
0 0 9
1 10 3 4 5
0 11 11
0 3 4
11 1 10 3 4 5
12 12 12
1 4 11
12 11 1 10 3 5
0 3 0
0 5 5
12 11 1 10 3

Table 5: Sample 2 Output File

At the beginning, there are three playlists. The first contains 3 and 5, the second contains 4, and the third contains 1. There is a limit 1 for Heartache, 3 for Roadtrip, 6 for Blissful. Each playlist can give at most 2 songs to EpicBlend in each category.

Before Events:

Playlist 1	Playlist 2	Playlist 3
song3 50 70 20 30 song5 10 50 50 50	song4 30 10 60 70	song1 100 99 0 50

	Heartache	Roadtrip	Blissful
	song1(99)	song4(60) song5(50) song3(20)	song4(70) song1(50) song5(50) song3(30)
From Playlist 1:	0	2	2
From Playlist 2:	0	1	1
From Playlist 3:	1	0	1

EpicBlend:	song1(100) song3(50) song4(30) song5(10)
------------	--

3.3.1 Event 1: ADD 9 2

Playlist 1	Playlist 2	Playlist 3
song3 50 70 20 30 song5 10 50 50 50	song4 30 10 60 70 song9 4 0 20 60	song1 100 99 0 50

	Heartache	Roadtrip	Blissful
	song1(99)	song4(60) song5(50) song3(20)	song4(70) song9(60) song1(50) song5(50) song3(30)
From Playlist 1:	0	2	2
From Playlist 2:	0	1	2
From Playlist 3:	1	0	1

EpicBlend:	song1(100) song3(50) song4(30) song5(10) song9(4)
------------	--

3.3.2 Event 2: ASK

EpicBlend:	song1(100) song3(50) song4(30) song5(10) song9(4)
------------	--

3.3.3 Event 3: ADD 10 2

Playlist 1	Playlist 2	Playlist 3
song3 50 70 20 30 song5 10 50 50 50	song4 30 10 60 70 song9 4 0 20 60 song10 100 80 0 90	song1 100 99 0 50

	Heartache	Roadtrip	Blissful
	song1(99)	song4(60) song5(50) song3(20)	song10(90) song4(70) song9(60) song1(50) song5(50) song3(30)
From Playlist 1:	0	2	2
From Playlist 2:	0	1	2
From Playlist 3:	1	0	1

Notice that Playlist 2 **can't** give more than two songs in the Blissful category. So, we removed Song 9.

EpicBlend:	song1(100) song10(100) song3(50) song4(30) song5(10) song9(4)
------------	---

3.3.4 Event 4: ASK

EpicBlend:	song1(100) song10(100) song3(50) song4(30) song5(10)
------------	--

3.3.5 Event 5: ADD 11 2

Playlist 1	Playlist 2	Playlist 3
song3 50 70 20 30 song5 10 50 50 50	song4 30 10 60 70 song9 4 0 20 60 song10 100 80 0 90 song11 300 0 80 80	song1 100 99 0 50

	Heartache	Roadtrip	Blissful
	song1(99)	song11(80) song4(60) song5(50) song3(20)	song10(90) song11(80) song4(70) song1(50) song5(50) song3(30)
From Playlist 1:	0	1	2
From Playlist 2:	0	2	2
From Playlist 3:	1	0	1

Notice that Playlist 2 **cannot** give more than two songs in the Blissful category. So, we removed Song 4. For the category Roadtrip, at most 3 songs can participate. Therefore, we removed worst scored song in the Roadtrip category.

EpicBlend:	song11(300) song1(100) song10(100) song3(50) song4(30) song5(10)
------------	---

3.3.6 Event 6: ASK

EpicBlend:	song11(300) song1(100) song10(100) song3(50) song4(30) song5(10)
------------	--

3.3.7 Event 7: ADD 12 2

Playlist 1	Playlist 2	Playlist 3
song3 50 70 20 30 song5 10 50 50 50	song4 30 10 60 70 song9 4 0 20 60 song10 100 80 0 90 song11 300 0 80 80 song12 10000 100 100 100	song1 100 99 0 50

	Heartache	Roadtrip	Blissful
	song12(100) song1(99)	song12(100) song11(80) song4(60) song5(50)	song12(100) song10(90) song11(80) song1(50) song5(50) song3(30)
From Playlist 1:	0	1	2
From Playlist 2:	1	2	2
From Playlist 3:	0	0	1

EpicBlend:	song12(10000) song11(300) song1(100) song10(100) song3(50) song4(30) song5(10)
------------	---

3.3.8 Event 8: ASK

EpicBlend:	song12(10000) song11(300) song1(100) song10(100) song3(50) song5(10)
------------	--

3.3.9 Event 9: REM 5 1

Playlist 1	Playlist 2	Playlist 3
song3 50 70 20 30 song5 10 50 50 50	song4 30 10 60 70 song9 4 0 20 60 song10 100 80 0 90 song11 300 0 80 80 song12 10000 100 100 100	song1 100 99 0 50

	Heartache	Roadtrip	Blissful
	song12(100)	song12(100) song11(80) song5(50) song3(20)	song12(100) song10(90) song1(50) song5(50) song3(30)
From Playlist 1:	0	1	1
From Playlist 2:	1	2	2
From Playlist 3:	0	0	1

EpicBlend:	song12(10000) song11(300) song1(100) song10(100) song3(50) song5(10)
------------	---

3.3.10 Event 10: ASK

EpicBlend:	song12(10000) song11(300) song1(100) song10(100) song3(50)
------------	--

4 Submission

The source files is collected under **/src**. Name of the main class is **Project3.java**.

The program can be compiled with the command below:

```
javac *.java
```

The input and output files can be in any folder. Design the code to accept the full path for file arguments. The program can be run with the command below:

```
java Project3 <song-file> <test-case-file> <output-file>
```