



## Roy and Trending Topics

Link submit: <https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/roy-and-trending-topics-1/description/>

**Solution:**

C++: <http://ideone.com/B2mNyy>

Python: <https://ideone.com/SaN6DA>

Java: <https://ideone.com/eKlvqd>

**Tóm tắt đề:** Cho bạn N topic với IDs và điểm phổ biến z-score. Bây giờ mỗi ngày z-score sẽ thay đổi do tương tác của người dùng, việc tính điểm thay đổi như sau:

1. Nếu là **"Post"** thì z-score sẽ tăng lên 50 điểm một post.
2. Nếu là **"Like"** thì z-score sẽ tăng lên 5 điểm một like.
3. Nếu là **"Comment"** thì z-score sẽ tăng lên 10 điểm một comment.
4. Nếu là **"Share"** thì z-score sẽ tăng lên 20 điểm một share.

Cho bạn danh sách những topic và các hoạt động của topic đó. Hãy tính toán và in ra 5 topic có điểm chênh lệch cao nhất, điểm chênh lệch = Old z-score – New z-score. Nếu 2 topic điểm bằng nhau thì bạn sẽ in ra topic nào có ID lớn hơn.

### Input

Dòng đầu tiên chứa số N là số lượng Topic cần xử lý. ( $1 \leq N \leq 10^6$ )

N dòng sau mỗi dòng là thông tin của 1 topic bao gồm các thông tin sau:

Topic **ID**, current z-score - **Z**, Posts - **P**, Likes - **L**, Comments - **C**, Shares – **S**

$1 \leq ID \leq 10^9$ ,  $0 \leq Z, P, L, C, S \leq 10^9$

### Output

In ra 5 topic có điểm thay đổi cao nhất và điểm mới của topic đó.

8	1003 200
1003 100 4 0 0 0	1002 300
1002 200 6 0 0 0	1001 400
1001 300 8 0 0 0	999 200
1004 100 3 0 0 0	1007 150
1005 200 3 0 0 0	

1006 300 5 0 0 0	
1007 100 3 0 0 0	
999 100 4 0 0 0	

**Giải thích:** Bảng điểm sau khi đã tính toán lại của các topic như sau:

Topic ID	Old z-score	New z-score	Change in z-score
1003	100	200	200-100 = 100
1002	200	300	300-200 = 100
1001	300	400	400-300 = 100
1004	100	150	150-100 = 50
1005	200	150	150-200 = -50
1006	300	250	250-300 = -50
1007	100	150	150-100 = 50
999	100	200	200-100 = 100

Như vậy ta thấy những topic có điểm Change in z-score lớn hơn sẽ được in ra trước, nếu bằng nhau thì ID lớn hơn sẽ in ra trước.

**Hướng dẫn giải:** Với mỗi topic bạn sẽ tính toán điểm chênh lệch = Old z-score – New z-score. Sau đó lưu vào một cấu trúc có 3 thành phần:

```
long long change_zscore;
int IDs;
long long new_zscore;
```

Viết thêm một hàm so sánh để phù hợp với yêu cầu đề bài. Sau khi bỏ vào hàng đợi ưu tiên thì sẽ lấy 5 giá trị đầu in ra là kết quả của bài toán.

Độ phức tạp:  $O(N\log N)$  với N là số lượng topic cần xử lý.