Samantha interviews many candidates from different colleges using coding challenges and contests. Write a query to print the contest_id, hacker_id, name, and the sums of total_submissions, total_accepted_submissions, total_views, and total_unique_views for each contest sorted by contest_id. Exclude the contest from the result if all four sums are **0**.

Note: A specific contest can be used to screen candidates at more than one college, but each college only holds **1** screening contest.

Input Format

The following tables hold interview data:

- Contests: The contest_id is the id of the contest, hacker_id is the id of the hacker who created the contest, and name is the name of the hacker.

Column	Type
contest_id	Integer
hacker_id	Integer
name	String

- Colleges: The college_id is the id of the college, and contest_id is the id of the contest that Samantha used to screen the candidates.

Column	Type
college_id	Integer
contest_id	Integer

- Challenges: The challenge_id is the id of the

```
stats
left join
(select
  challenge_id,
  sum(total_views) as sum_views,
  sum(total_unique_views) as sum_unique_views
  from view_stats group by challenge_id)
as views_sums
on challenges.challenge_id =
views_sums.challenge_id

-- group the information per-contest so that
everything is aggregated.
group by contests.contest_id, contests.hacker_id,
contests.name

-- HAVING works like WHERE, except over
aggregations, which is what we want here.
having (
  sum(submissions_sums.sum_submissions) +
  sum(submissions_sums.sum_accepted_submissions)
+
  sum(views_sums.sum_views) +
  sum(views_sums.sum_unique_views)
) > 0
order by contests.contest_id
```

Line: 45 Col: 29

Upload Code as File | Run Code | Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Your Output (stdout)

```
845 579 Rose 1987 580 1635 566
858 1053 Angela 703 160 1002 384
883 1055 Frank 1121 319 1217
```