Haskell Engineer Task

Suppose you have 2 files : authors.json and articles.json (attached in mail)

Your task is to create a command line Haskell program which produces a third file authors_with_articles.json defined as follows:

For each `author` in authors.json, add a field called `articles` where articles could be:

1. All articles of the given author (REQUIRED)
2. First 5 articles of the given author (OPTIONAL)
3. Articles of the given author which have a rating greater than 3 (OPTIONAL)

author.json is a list of `author` where each `author` has the following model:

```
{
  id: <integer> UNIQUE,
  name: <string>
}
```

articles.json is a list of `article` where each `article` has the following model:

```
{
  id : <integer> UNIQUE,
  author_id: <integer> FOREIGN KEY,
  title: <string>,
  rating: <integer>,
  created_id: <date>
}
```

Example

Suppose, authors.json is

```
[{ “id” : 1, “name”: “Harsh”},
 { “id” : 2, “name”: “Kriti”}]
```

articles.json is

```
[{ “id”: 1, “author_id”: 1, “title”: “my first article”, “rating”: 2, “created_at”: “2019-10-15” },
 { “id”: 2, “author_id”: 2, “title”: “my first article”, “rating”: 3, “created_at”: “2019-10-16” },
 { “id”: 3, “author_id”: 2, “title”: “my first article”, “rating”: 4, “created_at”: “2019-10-17” },
 { “id”: 4, “author_id”: 2, “title”: “my first article”, “rating”: 5, “created_at”: “2019-10-18” }]
```

Output: authors_with_articles.json should be (ignore formatting):

```
[{ “id” : 1, “name”: “Harsh”, “articles”: [
    { “id”: 1, “author_id”: 1, “title”: “my first article”, “rating”: 2, “created_at”: “2019-10-15” }]
 },
 { “id” : 2, “name”: “Kriti”, “articles”: [
   { “id”: 2, “author_id”: 2, “title”: “my first article”, “rating”: 3, “created_at”: “2019-10-16” },
   { “id”: 3, “author_id”: 2, “title”: “my first article”, “rating”: 4, “created_at”: “2019-10-17” },
   { “id”: 4, “author_id”: 2, “title”: “my first article”, “rating”: 5, “created_at”: “2019-10-18” }]
}]
```

Sample invocations

$ ./joinFiles authors.json articles.json (REQUIRED)
Output: Join each author with all their articles

$ ./joinFiles authors.json articles.json filter=first5 (OPTIONAL)
Output: Join each author with their first 5 articles (according to `createdAt`)

$ ./joinFiles authors.json articles.json filter=ratingGreater3 (OPTIONAL)
Output: Join each author with all their articles which have a `rating` greater than 3

$ ./joinFiles authors.json articles.json filter=first5 filter=ratingGreater3 (OPTIONAL)
Output: Join each author with their first 5 articles which have a `rating` greater than 3

---END---