Bang for Your Buck: Exercise

Write a point-free function to get the 3 top-rated meals <= a given price.

Given a maximum price and menu, return the 3 top-rated meals for that price or less.

Usage

```
// top 3 meals for $12 or less
const best3Meals = getTop3MealsFor(12, menu);

/*
[{
  name: 'Lamb Gyro',
  price: 11.86,
  rating: 4.9
}, {
  name: 'House Salad',
  price: 9.00,
  rating: 4.65
}, {
  name: 'Gigantus Fries',
  price: 11.86,
  rating: 4.5
}]
*/
```

Your solution must be point-free.

Notice how the `menu` parameter's supplied last, aligning with Ramda's "data-last" pattern. This lets you compose `getTop3MealsFor` with other functions to manipulate `menu` in different ways.

```
const firstPerfectMeal = pipe(
  getTop3MealsFor(20),
  filter(
    both(isVegetarian, isLactoseFree)
  ),
```

```
  head
);
```

This finds the best $20 or less vegetarian/lactose-free meal. All it needs is a `menu` parameter and off it goes!