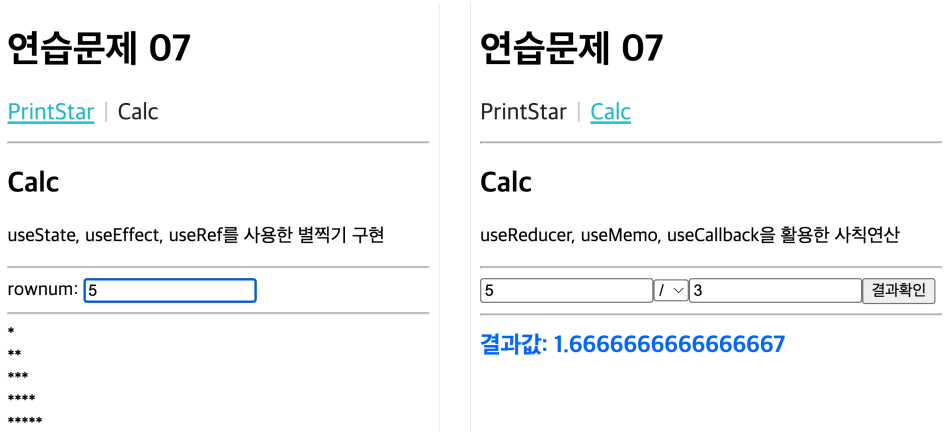
연습문제 07

다음의 요구사항을 충족하는 PrintStar.js, Calc.js 페이지를 만들어 App.js를 통해 화면에 표시하시오.



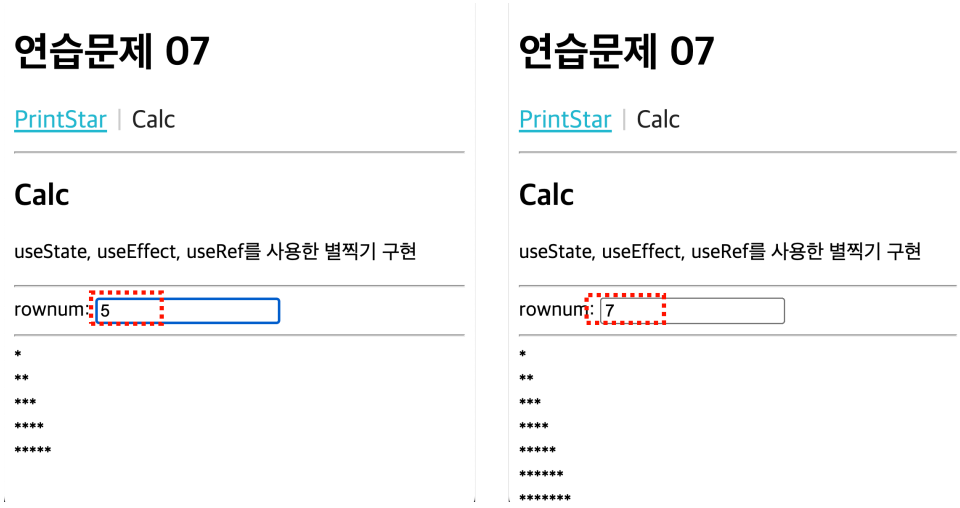
PrintStar.js

요구사항

1. `useState()`를 통해 관리되는 `rowNum` 상태변수를 정수형으로 생성하고 이 값을 `<input>`태그의 입력값이 변경될 때 갱신하시오.
2. `console`이라는 이름의 참조변수를 생성하여 이를 결과 출력을 위한 `<div>`태그에 연결하시오.
3. `rowNum`값이 변경되었을 때 `rowNum`만큼의 행을 갖는 별찍기 처리 로직을 구현하시오.
    - 별은 첫 번째 행에서는 1개, 2번째 행에서는 2개와 같이 하나씩 증가합니다.
    - 예를 들어 rowNum이 4일 경우 아래와 같이 표시됩니다.

```
*
**
***
****
```

실행결과



Calc.js

요구사항

1. 아래와 같은 구조의 입력 양식을 순차적으로 구성합니다.

```
input[type="text"][ref=x]
select[ref=exec]
input[type="text"][ref=y]
button[onclick=onButtonClick]
```

2. 드롭다운은 +,-,*,/를 선택할 수 있습니다.
3. `getResultValue()`라는 리듀서 함수에 의해 갱신되는 `resultValue`라는 상태값을 정의합니다.
4. `getResultValue()` 리듀서 함수는 액션값으로 `x`의 입력값, `y`의 입력값, `exec`의 선택항목을 전달받아 `exec`에 선택된 기호에 따라 적합한 사칙연산을 수행하고 그 결과를 반환합니다.
5. 버튼이 클릭되었을 때 `resultValue`를 갱신할 수 있는 이벤트 핸들러를 최적화 된 형태로 구현하세요.
6. 별도의 HTML DOM 요소를 정의하고 `resultValue`를 출력합니다.
7. 이 때 `resultValue`가 짝수면 `#f60`으로 처리하고 홀수이면 `#06f`의 색상값을 저장하는 `color` 상태변수를 만드세요.
8. `color` 색상값은 `reusltValue`가 이전과 같은 값일 경우 동작하지 않아야 합니다.
9. 도출된 `color`값을 결과 출력을 위한 DOM요소에서 style을 통한 글자 색상으로 적용하세요.

실행결과