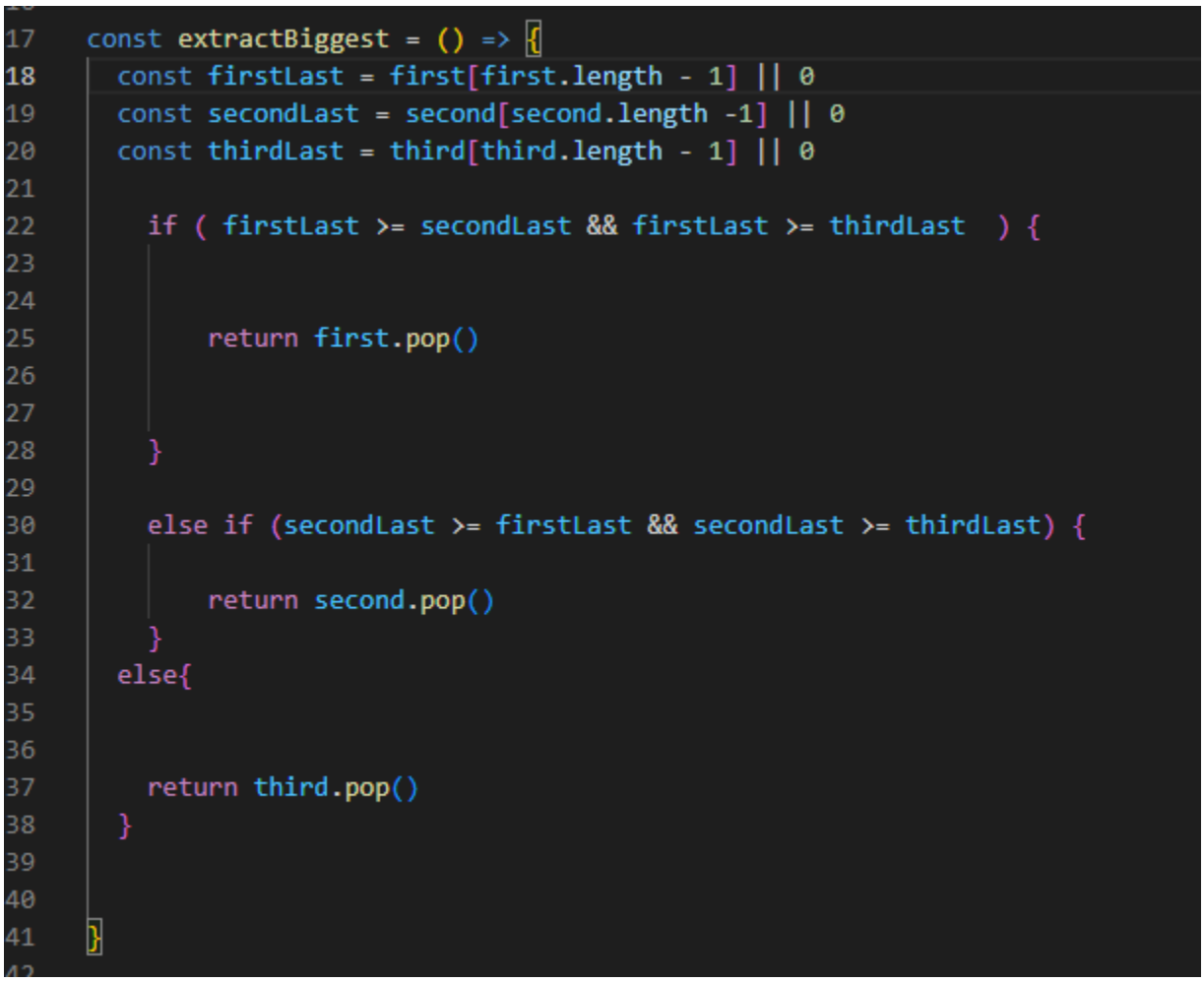
DWA_03.4 Knowledge Check_DWA3.1

---

1. **Please show how you applied a Markdown File to a piece of your code.**



```
const extractBiggest = () => {
  const firstLast = first[first.length - 1] || 0
  const secondLast = second[second.length -1] || 0
  const thirdLast = third[third.length - 1] || 0

    if ( firstLast >= secondLast && firstLast >= thirdLast  ) {


        return first.pop()


    }

    else if (secondLast >= firstLast && secondLast >= thirdLast) {

        return second.pop()
    }
  else{


    return third.pop()
  }


}
```

---

2. Please show how you applied JSDoc Comments to a piece of your code.

```
/**
createPreview - creates a preview element for a book with the given props
@param {Object} props - an object with properties of the book, including author, id, image, and title
@param {string} props.author - the id of the book's author in the authors object
@param {string} props.id - the id of the book
@param {string} props.image - the url of the book's cover image
@param {string} props.title - the title of the book
@returns {HTMLElement} - a button element with the book's preview
*/

export const createPreview = (props) => {

    const { author: authorId, id, image, title } = props

    let element = document.createElement('button')
    element.classList = 'preview'
    element.setAttribute('data-preview', id)

    element.innerHTML = /* html */ `
        <img
            class="preview__image"
            src="${image}"
        />

        <div class="preview__info">
            <h3 class="preview__title">${title}</h3>
            <div class="preview__author">${authors[authorId]}</div>
        </div>
    `
    return element
}
```

---

3. Please show how you applied the @ts-check annotation to a piece of your code.

```
// @ts-check

const data = {
  lists: [
    ['first', [15, 11, 13, 7, 5]],
    ['second', [2, 6, 8, 4, 14, 12, 10]],
    ['third', [9, 3, 1]],
  ],
};

// Only edit below

const first = data.lists[0][1] || [];
const second = data.lists[1][1] || [];
const third = data.lists[2][1] || [];

const result = [];

const extractBiggest = () => {
  const firstLast = first[first.length - 1] || 0;
  const secondLast = second[second.length - 1] || 0;
  const thirdLast = third[third.length - 1] || 0;

  if (firstLast >= secondLast && firstLast >= thirdLast) {
    return first.pop();
  } else if (secondLast >= firstLast && secondLast >= thirdLast) {
    return second.pop();
  } else {
    return third.pop();
  }
};
```

---

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.