Olivia Axiuk, Julia Axiuk
CART 263
Task 7-8

1. The array has 150 objects, each object has a:
    a. petal Length
    b. petalWidth
    c. sepalLength
    d. sepalWdith
    e. Species

2.

```
 let possibleColor = ["#5d3fd3", "#a73fd3", "#d33fb5", "#d35d3f",
"#d3a73f"]
    const irisesWithColors = irises_original.map(iris => {
      return {
        ...iris,
        color: possibleColor[Math.floor(Math.random() *
possibleColor.length)]
      };
    });
```

3.

```
const filteredIrises = irisesWithColors.filter(iris => iris.sepalWidth <
4);
```

4.

```
const totalPetalLength = irisesWithColors.reduce((sum, iris) => sum +
iris.petalLength, 0)
    console.log(totalPetalLength)
    console.log(irisesWithColors.length)
    //Total Petal Length: 563.70000
    // Average Petal Length: 3.758
```

5.

```
const foundIris = irisesWithColors.find(
    function (el) {
      return (el.petalWidth > 1.0)
```

```
      })
    console.log(foundIris)
    //foundIris: {sepalLength: 7, sepalWidth: 3.2, petalLength: 4.7,
petalWidth: 1.4, species: 'versicolor', …}
    // color
    // :
    // "#5d3fd3"
    // petalLength
    // :
    // 4.7
    // petalWidth
    // :
    // 1.4
    // sepalLength
    // :
    // 7
    // sepalWidth
    // :
    // 3.2
    // species
    // :
    // "versicolor"
    // [[Prototype]]
    // :
    // Object
```

6.

```
const hasPetalLengthLessthan10 = irisesWithColors.some(
      function (el) {
        return (el.petalLength > 10)
      }
    )
    console.log(hasPetalLengthLessthan10)
    //answer: False
```

7.

```
  //part07
    const petalEqualtoFour = irisesWithColors.some(
      function (el) {
```

```
        return (el.petalLength == 4.2)
      }
    )
    console.log(petalEqualtoFour)
    //answer: true
```

8.

```
//part 08

    const everyPetalWidth3 = irisesWithColors.every(
      function (el) {
        return (el.petalWidth < 3)
      });
    console.log(everyPetalWidth3);
    //answer: true
```

9.

```
//part 09
    const everySepalWidth = irisesWithColors.every(
      function (el) {
        return (el.sepalWidth > 1.2)
      });
    console.log(everySepalWidth);
    //answer: true
```

10.

```
 //part 10


    const irisesWithColorsSorted = irisesWithColors.toSorted((a, b) =>
a.petalWidth - b.petalWidth);
    console.log(irisesWithColorsSorted);
```

11.

For our visualization, we decided to create abstract flowers in a flower box, with their size, shape and color depending on the information in the JSON, and their color being taken from the ‘irisesWithColors’ array. The Irises are drawn based on a function in the constructor,

with their X and Y changing and moving to the ‘bucket’ when they are ‘picked’ using a click function. As they are clicked, a price tag appears on the side, showing the price of the total flowers picked in cents. This is done also in the click function within the constructor.

We also implemented some simple HTML and CSS for the visualization, adding an extra header to which the ‘price tag’ could be appended to, and using the CSS to place, color and shape the different elements for a (rather abstract) spring theme.