Solution Review: Max with Nested Functions

In the following lesson, we will go over the solution for the challenge: Max with Nested Functions.

We'll cover the following

Task

In this challenge, you had to create a nested function `max` which would help its parent function `mainMax` to compute the maximum of three numbers.

Solution

A skeleton of the `mainMax` function was already provided for you. Let's look it over.

```
int mainMax(int a, int b, int c) {

}
```

`mainMax` takes three parameters of type `int` and returns a value of type `int`.

Let's go over the step-by-step process for writing the `max` function.

- `max` is intended to break down the bigger problem into a smaller one. While `mainMax` returns the maximum of three numbers, `max` returns the maximum of two of them. This means that it will take two parameters of type `int` and return the greater of the two. To find the maximum of two numbers, a simple `if-else` expression can be used.

```
int max(int x, int y) {
    if(x > y){
      return x;
    } else{
      return y;
    }
```

```
}
```

- As for the return value of `mainMax`, we simply needed to call the `max` function. The first argument will be one of the three numbers passed to `mainMax` and the second argument will be the maximum of the remaining two. To get the second argument, we will use the `max` function again as it returns the maximum of two numbers.

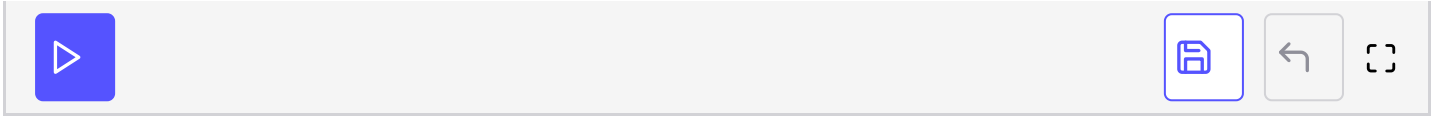
```
return max(a,max(b,c));
```

You can find the complete solution below:

> You were required to write the code given from **line 1** to **line 10**.

```
int mainMax(int a, int b, int c) {
  int max(int x, int y) {
    if(x > y){
      return x;
    } else{
      return y;
    }
  }
  return max(a,max(b,c));
}

main() {
  var result = mainMax(1,9,5);
  print(result);
}
```



Let’s end this chapter with a quiz to test what you have learned so far.