Also, make sure you know how to declare various forms of array. Here is another array type of question I can ask you.

Consider the following JavaScript code, which you should assume compiles and runs without errors.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE=JavaScript>

var myList = new Array(6, 3, 9, 5);

function doPartA(i, j) {
      var temp;
      temp = myList[i];
      myList[i] = myList[j];
      myList[j] = temp;
}


function doPartB(s) {
      for (k=s+1; k < 4; k++) {
              if (myList[k] < myList[s])
                      s = k;
      }
      return s;
}

function doPartC() {
      for (m=0; m < 4; m++) {
            x = doPartB(m);
            doPartA(m, x);
      }
}

function showResult() {
      alert(" "+myList[0]+" "+myList[1]+" "+myList[2]+" "+myList[3]);
}

</SCRIPT>
</HEAD>
<BODY>
<INPUT type="button" value="Click Me" onclick="doPartC()">
</BODY>
</HTML>
```

In the answer box provided, show the contents of myList after I click the “Click Me” button. You are to show your incremental work for full credit in the space provided below:

Instead of showing you just the results, I put a trace here to help you understand how the arrays changed. This trace allows you to follow the code and see what happen. Hope this is helpful.

IN PART C

```
m: 0
IN PART B
INITIAL s: 0
k: 1
Change s to be k: 1
s: 1
k: 2
s: 1
k: 3
s: 1
DONE WITH PART B
x: 1
IN PART A, SWAPPING i and j
i: 0, j: 1
RESULTING ARRAY
3 6 9 5

m: 1
IN PART B
INITIAL s: 1
k: 2
s: 1
k: 3
Change s to be k: 3
s: 3
DONE WITH PART B
x: 3
IN PART A, SWAPPING i and j
i: 1, j: 3
RESULTING ARRAY
3 5 9 6

m: 2
IN PART B
INITIAL s: 2
k: 3
Change s to be k: 3
s: 3
DONE WITH PART B
x: 3
IN PART A, SWAPPING i and j
i: 2, j: 3
RESULTING ARRAY
3 5 6 9

m: 3
IN PART B
INITIAL s: 3
DONE WITH PART B
x: 3
IN PART A, SWAPPING i and j
i: 3, j: 3
RESULTING ARRAY
3 5 6 9
```