CPSC 1045 In Class worksheet: Arrays

Part 1:
The following table represents the array yogurt.

	0	1	2	3
yogurt	"Green Apple"	"Banana"	"Chocolate"	"Strawberry"

1. Write the JavaScript code necessary to create the above array.

var yogurt = ["Green Apple", "Banana", "Chocolate", "Strawberry"];

2. Expression evaluation

    a. What is the value of `s1` after the following code.
    ```
    var s1 = yogurt[2] + " is not very tasty";
    ```
    Answer: "Chocolate is not very tasty"

    b. What is the value of s2 after the following code:
    ```
    yogurt[0] = "Cheese";
    var s2 = yogurt[0] + " yogurt is tasty!";
    ```
    Answer: "Cheese yogurt is tasty!"

    c. What is the values of yogurt after the following code:
    ```
    yogurt[0] = yogurt[2] + yogurt[2+1];
    yogurt[2] = "Mint";
    ```
    Answer: ["ChoclateStrawberry", "Banana", "Mint", "Strawberry"]

3. Write a function called **findC** that accepts an array of strings as a parameter and returns all the strings that begin with "C" in a new array.

```
function findC(stringArray){
   "use strict";
    var cArray = [];
    for(var i = 0; i < stringArray.length; i = i +1){
          if(stringArray[i].substring(0,1) === "C"  ){
              cArray.push(stringArray[i]);
          }
    }
    return cArray;
}
```

What is the value of outArr?
var outArr = findC(yogurt);
Answer: ["ChocolateStrawberry"]

Part 2:
The following table represents the array yogurt.

	0	1	2	3	4
factors	2	2	3	5	5

1. Write the JavaScript code necessary to create the above array.

var factors=[2,2,3,5,5];

2. Expression Evaluation
    a. What is the value of diff after the following code executes.

    var diff = factors[4]-factors[2];

    Answer: 2

3. Write a function called **multArray** that takes in an array of numbers as a parameter and returns the product of all the numbers in the array.

```
function MultArray(numArray){
   "use strict";
   var product = 1;
   for(var i = 0; i < numArray.length; i = i +1){
       product *=  numArray[i];
   }
   return product;
}
```

With your function evaluate the following expression

var prod = multArray(factors);

prod is now 300

4. Write a function called uniqueNumbers that takes in an array of number as a parameter and returns an array with all the unique values in the array.

With your function evaluate the following expression and determine the value of uniqueFactors.
var uniqueFactors = uniqueNumbers(factors);

```
function uniqueNumbers(numArray){
   "use strict";

   var uniqueList = [];

   for(var i =0; i < numArray; i = i +1){
      if(uniqueList.indexOf(numArray[i]) === -1 ){
            uniqueList.push(numArray[i]);
      }
   }
   return uniqueList;
}
```

Part 3:

Write a function called oddNumbers, that takes in a parameter N and returns the first N odd numbers in an array starting from 3.

So
var oddList = oddNumbers(3)

oddList will have the value of [3, 5, 7]

```
function oddNumber(N){
   "use strict";
   var oddNums = [];
   var currentNum = 3;
   for(var i =0; i < N; i = i +1){
        oddNums.push(currentNum);
        currentNum = currentNum +2;
   }
   return oddNums;
}
```