Use your answers to demonstrate your familiarity with *modern* Javascript concepts. Feel free to employ features such as regular expression, map, reduce, filter, recursion, etc. as you see fit. You may even choose to provide alternate answers in a different language. I urge you though to provide elegant solutions that are no more than a few lines long. If you find yourself using nested loops or conditionals, you're probably on the wrong track.

You are free to use the Internet for research, but the answers you provide should ultimately be your own. Sending your responses via a repl.it link (or similar) would be helpful.

**** Question 1**

Given an array of strings such as

```
["A", "B", "C"]
```

return an array of objects such as

```
[ {"name": "A"}, {"name": "B"}, {"name": "C"} ]
```

**** Question 2**

Given an array of string, such as

```
["A", "B", "C", "Z", "Q"]
```

return an object such as

```
{ "A": 0,  "B": 1,  "C": 2,  "Z" : 3,  "Q": 4 }
```

The numbers here are the positions of the string in the input array.

**** Question 3**

Suppose the tree

```
              1
             / \
            2   4
           /   / \
          3   5   6
```

is represented by the object

```
tree = {
 value: 1,
 children: [ { value: 2,
               children: [ { value: 3,
                             children: []
                           }
                         ]
             },

             { value: 4,
               children: [ { value: 5,
                             children: []
                           },
                           {
                             value: 6,
                             children: []
                           }
                         ]
             }
           ]
}
```

How would you total up the "value" of all the nodes. For example, here the answer would be 1 + 2 + 3 + 4 + 5 + 6 = 21.

**** Question 4**

Given an array of words, pick out only those words that have two or more vowels in them. For the purposes of this question, a vowel is one of the letters a, e, i, o, u.

For example, given

```
["dog", "cat", "mouse", "sky", "eleven"]
```

return

```
["mouse", "eleven"]
```

**** Question 5**

Given the same input as above, place parentheses around the vowels. For example:

```
["d(o)g",  "c(a)t",  "m(ou)s(e)",  "sky",  "(e)l(e)v(e)n"]
```

Notice how consecutive vowels are grouped. That is, "m(ou)s(e)" is correct; "m(o)(u)s(e)" is incorrect.