22

23 }

return sum * multiplier

Your Solutions

Run Code

Our Solution(s) Run Code

```
Solution 1
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   package main
5 type SpecialArray []interface{}
^7\, // O(n) time | O(d) space - where n is the total number of elements i
   // including sub-elements, and d is the greatest depth of "special" a
9 func ProductSum(array SpecialArray) int {
    return helper(array, 1)
10
11 }
12
13 func helper(array SpecialArray, multiplier int) int {
     sum := 0
14
     for _, el := range array {
      if cast, ok := el.(SpecialArray); ok {
16
17
         sum += helper(cast, multiplier+1)
       } else if cast, ok := el.(int); ok {
18
19
         sum += cast
20
21
     }
```

```
package main

type SpecialArray []interface{}

// Tip: Each element of `array` can either be cast

// to `SpecialArray` or `int`.

func ProductSum(array []interface{}) int {
    // Write your code here.
    return -1
}
```

 Our Tests
 Custom Output
 Submit Code

Run or submit code when you're ready.

The Contract Contract
