

```
1 #Mod 10 Tutorial
2 #Dr. Burkman
3
4 $input_counter = 0
5 $myArray = @()
6
7 $myArray = @('seven', '7', '120', 'monkey', 'cat', 'the', 'heck', '3.1',
... '-1384', 'banana')
8
9
10
11 #This function randomly inserts whatever is passed to it into the string.
12 function RandoInsert($myArray, $thingBeingInserted)
13 {
14     $tempArray = @()
15     $indexPosition = Get-Random -Minimum 0 -Maximum ($myArray.Length + 1)
16     if($indexPosition -eq 0)
17     {
18         $tempArray += $thingBeingInserted
19         for($i = 0;$i -lt $myArray.Length;$i++)
20         {
21             $tempArray += $myArray[$i]
22         }
23         return $tempArray
24     }
25     elseif($indexPosition -ge $myArray.Length)
26     {
27         for($i = 0;$i -lt $myArray.Length;$i++)
28         {
29             $tempArray += $myArray[$i]
30         }
31         $tempArray += $thingBeingInserted
32         return $tempArray
33     }
34     else
35     {
36         for($i = 0;$i -lt $indexPosition;$i++)
37         {
38             $tempArray += $myArray[$i]
39         }
40         $tempArray += $thingBeingInserted
41         for($i = $indexPosition;$i -lt $myArray.Length;$i++)
42         {
43             $tempArray += $myArray[$i]
44         }
45         return $tempArray
46     }
47 }
48
49
50
```

```
51
52
53 clear
54 Read-Host "Press enter to move from task to task"
55 clear
56
57
58 <#
59
60 #Task 1
61 #Fill the list with user input
62 clear
63 Write-Host "Task 1`r`n"
64 Write-Host ("Create an array filled with some words and numbers")
65 while ($input_counter -lt 10)
66 {
67     $user_input = Read-Host "Please enter a number or word"
68     $myArray += $user_input
69     $input_counter += 1
70 }
71
72 #>
73
74 #Task 2
75 #Check the list to ensure it has 10 items
76 clear
77 Write-Host "Task 2`r`n"
78 $myArrayLength = $myArray.Length
79 $boolCheck = $myArrayLength -eq 10
80 Write-Host ("This array has $myArrayLength items.  $boolCheck")
81 Read-Host
82
83 #Task 3
84 #Print out the array
85 clear
86 Write-Host "Task 3`r`n"
87 Write-Host "This is the array:"
88 Write-Host $myArray
89 Read-Host
90
91 #Task 4
92 #Swap the first item with the last item in the array and print it out
93 #clear
94 Write-Host "Task 4`r`n"
95 $firstThing = $myArray[0]
96 $myArray[0] = $myArray[-1]
97 $myArray[-1] = $firstThing
98 Write-Host "This is the array after swapping the first and last items:"
99 write-host $myArray
100 Read-Host
101
```

```
102 #Task 5
103 #Print the first 3 and last 3 items in the array
104 #clear
105 Write-Host "Task 5`r`n"
106 Write-Host("These are the first three and last three items in the array:")
107 write-host ($myArray[0..2], $myArray[-3..-1])
108 Read-Host
109
110 #Task 6
111 #Loop through and print out the individual items in the array
112 clear
113 Write-Host "Task 6`r`n"
114 foreach ($i in $myArray)
115 {
116     Write-Host $i
117 }
118 Read-Host
119
120 #Task 7
121 #Check to see if cat is in my array and let the user know
122 clear
123 Write-Host "Task 7`r`n"
124 if ($myArray.Contains("cat"))
125 {
126     Write-Host "There is a cat in my array"
127 }
128 else
129 {
130     Write-Host "There is no cat in my array"
131 }
132 Read-Host
133
134 #Task 8
135 #Get the name of a Marvel character from the user and pass it to the random
... insert function
136 clear
137 Write-Host "Task 8`r`n"
138 $userHero = Read-Host "Please insert the name of a Marvel character"
139 $myArray = RandoInsert -myArray $myArray -thingBeingInserted $userHero
140 Write-Host ("The array is now: $myArray")
141 Read-Host
142
143 #Task 9
144 #Print out the index position of the hero, then the array
145 clear
146 Write-Host "Task 9`r`n"
147 Write-Host $userHero "is at position" $myArray.IndexOf($userHero) "in the
... array"
148 Read-Host
149
150
```

```
151 #Task 10
152 #Copy integers to new array, sort, print
153 clear
154 Write-Host "Task 10`r`n"
155 $intArray = @()
156 foreach ($i in $myArray)
157 {
158     try
159     {
160         $diff = $i - [int]$i
161         $i = [int]$i
162         if ($diff -ne 0)
163         {
164             continue
165         }
166         else
167         {
168             $intArray += $i
169         }
170     }
171     catch
172     {
173         continue
174     }
175 }
176
177 $intArray = $intArray | sort
178 Write-Host "The integers in the original array, sorted, are: $intArray"
179
180
181 #Task 11
182 #Unpack an array of arrays
183 clear
184 Write-Host "Task 11`r`n"
185 $a= @(1,2,3)
186 $b = @("a","b","c")
187 $c = @(1.1, "house", 7)
188 $d = @($a, $b, $c)
189
190 foreach ($i in $d)
191 {
192     foreach ($j in $i)
193     {
194         write $j
195     }
196 }
197
198 Read-Host -Prompt "`r`nPress enter to close the program"
199
```