## Boolean Expression Evaluation

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>True or 4 == 3</td>
<td>True</td>
<td>boolean</td>
</tr>
<tr>
<td>3 &lt; 4 &lt;= 10</td>
<td>True</td>
<td>boolean</td>
</tr>
<tr>
<td>not(4 == 5) or 6 &gt; 5 and 4 &gt; 5</td>
<td>False</td>
<td>boolean</td>
</tr>
<tr>
<td>6 + 2 &gt; 10 &lt; 4 * 2</td>
<td>False</td>
<td>boolean</td>
</tr>
<tr>
<td>True or False</td>
<td>True</td>
<td>boolean</td>
</tr>
<tr>
<td>True and False</td>
<td>False</td>
<td>boolean</td>
</tr>
<tr>
<td>not(True) or True or not(False)</td>
<td>True</td>
<td>boolean</td>
</tr>
<tr>
<td>7 + 3 / 2 &gt;= 8</td>
<td>True</td>
<td>boolean</td>
</tr>
</tbody>
</table>
Code Tracing

What is printed when the following code is executed?

```python
l = ["open", "close", "in", "out", "up", "down"]
for i in range(0,6,2):
    print l[i]
```

Answers:
open
in
up
List Practice

What will be the output of the following sequence of code?

```python
mylist = ["Hello", 88.5, 100, "cs190"]
print len(mylist)  # 4
print mylist[0]    # Hello
print mylist[len(mylist)]  # Error
print mylist[2:]    # [100, “cs190”]
print mylist[0:2]   # [“Hello”, 88.5]
mylist.append(140)
print mylist        # [“Hello,88.5,100,”cs190”,140]
mylist[0] = 4
mylist[3] = 200
print mylist        # [4, 88.5, 100, 200, 140]
mylist.sort()
print mylist        # [4, 88.5, 100, 140, 200]
mylist.reverse()
print mylist        # [200, 140, 100, 88.5, 4]
```
def fun1(x):
    print "Fun1 x:", x
    return x * 2
print "Start"
y = fun1(10)
if (5 > y ):
    print y
elif ( 15 > y ):
    print y + 10
elif (25 > y ):
    print y + 100
elif (35 > y):
    print y + 1000
else:
    print y + 10000
print "End"

This code utilizes if statements
Which won't be on the exam.
Write a function called two which takes a list of numbers as an input parameter. This function should return True if 2 appears as either the first or last element in the array. The list will be length 1 or more. Examples of function calls:

two([1, 2, 2]) → True
two([2, 1, 2, 3]) → True
two([13, 6, 1, 2, 3]) → False
Function Writing Answer

def two(mynums):
    last_idx = len(mynums) - 1
    return (mynums[0] == 2) or (mynums[last_idx] == 2)
def stringTimes(s, i):
    newstr = ''
    for x in range(i):
        newstr = newstr + s
    return newstr
def sumEvens(n):
    sum = 0
    for i in range(0, n+1, 2):
        sum = sum + i
    return sum

Note: The tricky part of this function is making sure n is included if it is even. The crucial bit is the 2nd parameter to the range function.