Problem Solving Methodology in IT (COMP1001)

Solution to the midterm test for Group 1

16 October 2018

1. [Weight = 1] The smallest is obtained by setting the integer and fractional parts to 0. Similarly, the second smallest is obtained by setting the integer part to 0 and the least significant digit in the fractional part to 1. The next two smallest numbers are done similarly.

   - **0.0 ($0.0000000000_{two} = 0_{ten}$)**
   - **0.0009765625 ($0.0000000001_{two} = 1/1024_{ten} = 0.0009765625_{ten}$)**
   - **0.001953125($0.0000000010_{two} = 2/1024_{ten} = 0.001953125_{ten}$)**
   - **0.0029296875 ($0.0000000011_{two} = 3/1024_{ten} = 0.0029296875_{ten}$)**
   - 0.00390625 ($0.0000000100_{two} = 4/1024_{ten}$)
   - …

2. [Weight = 1] The first line is the output of the first *print()* which prints out only *outer* and *middle*. The next two lines are printed by the second *print()* which prints out all three values. This pattern repeats with the second value of *middle* and so on until all the values of *middle* are exhausted. Note that the last line is repeated twice, because the second and third *print()*s print out the same value. The next set of values is the same except that *outer* is equal to 2.

```
1 3
1 3 6
1 3 7
1 4
1 4 6
1 4 7
1 5
1 5 6
1 5 7
1 5 7
2 3
2 3 6
2 3 7
2 4
2 4 6
2 4 7
2 5
2 5 6
2 5 7
2 5 7
```

3. [Weight = 1]

```
Function real_bin2Dec(r)
    Input: r is a string representing a real number in binary
    (e.g., "101.001").
    Output: return the decimal value of the binary number in r.

    result ← 0.
    posRP ← posRadixPt(r)

    for i in [0, …, posRP-1] do
        result ← result + 2**(posRP-i-1)*r[i]

    for i in [posRP+1,len(r)-1] do
        result ← result + 2**(-(i-posRP))*r[i]

    return result
```

4. [Weight = 1] See the Gp1-Q4.py

5. [Weight = 1] See the Gp1-Q5.py