

# Programming Assignment # 1

Name: Mark Fastner

Code:

Task1:

```
Task 1.py x Task 2.py x Task 3.py x
2 def gcd(a, b):
3     """returns the gcd of integers a and b
4     INPUT:
5         a, b - integers
6     OUTPUT:
7         the gcd as an int
8
9     """
10    r = 0
11
12    while True:
13        r = a % b
14        if r == 0:
15            return b
16            break
17
18        a = b
19        b = r
20
21    pass
```

```
# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    """TESTER CELL"""

    print("Testing gcd(101, 4620) = ", gcd(101, 4620))
    print("Expected: 1")
    print("Testing gcd(2349, 36) = ", gcd(2349, 36))
    print("Expected: 9")
```

Output:

```
"/Users/mark/Documents/School/spring 2021/229 lab/pycharm stuff/venv/bin/python" "/Users/mark/Documents/School/spring 2021/229 lab/pycha
Testing gcd(101, 4620) = 1
Expected: 1
Testing gcd(2349, 36) = 9
Expected: 9

Process finished with exit code 0
```

## Task 2:

```
def binary_to_dec(bitstring):
    newstring = bitstring.replace(" ", "")
    sum = 0
    power = len(newstring) - 1
    for element in newstring:
        sum += int(element) * 2**power
        power = power - 1
    return sum
"""
converts the given bitstring into the decimal representation of the number.
INPUT:
    bitstring - a string containing 0's and 1's that is the binary representation of the number
OUTPUT:
    the decimal representation of the bitstring as an integer
"""
pass
```

```
if __name__ == '__main__':
    """TESTER CELL"""

    print("Testing binary_to_dec(\"11 0011 1001\") = ", binary_to_dec("11 0011 1001")) # your solution should support strings with spaces
    print("Expected: 825")
```

## Output:

```
Task 2
"/Users/mark/Documents/School/spring 2021/229 lab/pycharm stuff/venv/bin/python" "/Users/mark/Documents/School/spring 2021/229 lab/pycharm stuff/venv/Task 2.py"
Testing binary_to_dec("11 0011 1001") = 825
Expected: 825

Process finished with exit code 0
```

### Task 3:

```
1  """ Part I: The Rectangle class """
2  class Rectangle:
3
4      #FIXME: implement the class here
5      def __init__(self):
6          self.width = 1
7          self.height = 1
8
9
10     def _init_(self, nwidth, nheight):
11         self.width = nwidth
12         self.height = nheight
13
14     def getArea(self):
15         return self.width * self.height
16
17     def getPerimeter(self):
18         return 2 * (self.width + self.height)
19
```

```
19
20 """Part II: The tester"""
21
22 #FIXME: Implement your tester in this cell
23 r1 = Rectangle(4, 40)
24 r2 = Rectangle(3.5, 35.9)
25
26 print("First Rectangle   Width:", r1.width, "Height:", r1.height, "Area:", r1.getArea(), "Perimeter:", r1.getPerimeter());
27 print("Second Rectangle  Width:", r2.width, "Height:", r2.height, "Area:", r2.getArea(), "Perimeter:", r2.getPerimeter());
```

### Output:

```
Task 3
"/Users/mark/Documents/School/spring 2021/229 lab/pycharm stuff/venv/bin/python" "/Users/mark/Documents/School/spring 2021/229 lab/pycharm stuff/venv/Task 3.py"
First Rectangle   Width: 4 Height: 40 Area: 160 Perimeter: 88
Second Rectangle  Width: 3.5 Height: 35.9 Area: 125.64999999999999 Perimeter: 78.8

Process finished with exit code 0
```