Instructions: This is a closed book, 80 minute exam worth a total of **30 marks**. You may only have a calculator and a pencil/eraser with you. Please remove any study aids, cell phones, etc from your desk. **Write legibly** – if we cannot read your writing, we cannot mark it.

(1) Convert the decimal number 404 to binary, then convert the binary number 0b110110101 to decimal. You must show your work in both cases to get full marks!

(2) Simplify the following logical expression. You must show your work to get credit:

\[
(\neg (\neg \text{true}) \land (\neg \text{false} \lor \neg \text{false})) \lor (\text{true} \land \neg \text{false} \lor \text{false})
\]
(4) What does the code to the right print?

```python
def q3():
    a = 10
    b = 2
    c = a - b * a
    if(c < a):
        print(a)
    else:
        print(c)
q3()
```

(5) What does the code to the right print?

```python
def q5():
    a = 0
    while a < 10:
        print(a)
        if a % 2 == 0:
            a = a + 1
        elif a == 6:
            a = a - 1
        else:
            a = a * 2
        print(a)
q5()
```

(6) What does the code to the right print?

```python
def q6(x, y, z):
    for i in range(x, z):
        w = y
        while w > x:
            print(i * w)
            w = w - 1
q6(1, 4, 3)
```
(7) What does the code to the right print?

```python
def q7(a, b, x):
    for i in range(a, b):
        out = x
        for j in range(i, 4):
            out = out + '!
        while out != x + '!!!':
            print(out)
            out = out + '!
q7(1, 5, "meow")
```

(8) Circle all the mistakes in the following code, and write out the correct code in the space provided (use comments to describe your changes if necessary). The correct code should define a function that takes a number argument, n, and returns the product of all numbers from 1 to n.

E.g. `product(3) → 1*2*3 = 6`

```python
product(n):
    result = 0
    for i in range(1, n):
        result = result * x
    return product
```
Create a function called `sumOfSquares()` that, given a number n as input, will calculate the sum of the squares of all numbers between 1 and n.

E.g. `sumOfSquares(4)`: \(1^2 + 2^2 + 3^2 + 4^2 = 1 + 4 + 9 + 16 = 30\)

(For bonus practice try doing this without using the multiplication operator!!)

```python
def sumOfSquares(n):
    total = 0
    for i in range(1, n + 1):
        total = total + (i * i)
    return total

def sumOfSquares_noMult(n):
    total = 0
    for i in range(1, n + 1):
        total += square(i)
    return total

def square(x):
    total = 0
    for i in range(0, x):
        total += x
    return total
```