1. (14 points) Evaluate the following simple expressions (as Python would).

(a) \(2 + 3 \times 4\)

(b) \(10 / 4\)

(c) \(10 \times 1.0\)

(d) \(10.0 / 4\)

(e) "Beseech"[5]

(f) "Beseech"[-2]

(g) "Beseech"[2:5]

2. (12 points) Evaluate the following logical expressions (as Python would).

(a) \(10 \leq 15 \leq 20\)

(b) \(17.5 > 17.5\)

(c) "ABC" == "abc"

(d) "abcdefg"[3] == "d"

(e) not True or False

(f) ord( "?" ) == 126 and ord( "?" ) != 126
3. (10 points) Consider the following Python code:

\[
\begin{align*}
x & = 4 \\
y & = 10 \\
\text{if } x > y: & \quad x = x + 1 \\
& \quad y = x \times y \\
\text{elif } x < y: & \quad x = x - 1 \\
& \quad y = x \times 2 \\
\text{else:} & \quad x = y \\
& \quad y = x
\end{align*}
\]

After running this code, what values would \( x \) and \( y \) have?

4. (14 points) Draw a stack diagram for the following code.

Remember to include all stack frames, as well as all defined functions and variables.

```python
def foo(u,v):
    sum = u+v
    prod = u*v
    ans = bar(sum, prod)
    return ans

def bar(x,y):
    z = 10*x
    return y+z

def baz(n):
    m = foo(n+1, n-1)
    return n + m

a = baz(3)
```
5. *(15 points)* What does the following function do?

Dont just list the steps it takes. Tell me what its purpose is!

You may assume that \(x\) and \(y\) are both integers.

```python
def mystery(x,y):
    """What do I do?"
    difference = x - y
    if difference > 0:
        return x
    else:
        return y
```

6. *(15 points)* What does the following function do?

Dont just list the steps it takes. Tell me what its purpose is!

You may assume that `string` is a string, and that `element` is a single-character string.

```python
def stringChecker(string, element):
    """What do I do?"
    if string == ":
        return False
    elif string[0] == element:
        return True
    else:
        return stringChecker(string[1:], element)
```
7. (20 points) Use recursion to write a function isEven(x)

The function isEven(x) should return True if x is even. Otherwise, it should return False. You may assume x is a non-negative integer.

You may find the following information useful.

1. 0 is even
2. 1 is odd
3. In general, an integer is even if the integer two numbers before it is also even.

NOTE You may not use the % operator when writing this function.