1. For each of the following code snippets, enter the final value for $x$ on the line [2 points]

a. 
```
x = 36
if x < 3:
    x = 3
else:
    x = 9
if x < 27:
    x = 27
```

b. 
```
x = 36
if x < 3:
    x = 3
elif x < 9:
    x = 9
elif x < 27:
    x = 27
```

2. We often need to restrict values to an allowable range. Write a function named `clamp` that takes 3 numeric arguments: a value, a minimum and a maximum. If the value is between the minimum and maximum, inclusive, it should return the value, if it less than minimum it should return minimum, if it is greater than maximum it should return maximum. Your function must use an if statement. You do not need to include comments or docstrings. Some examples are below. [4 points]

```python
>>> clamp(0.5, -1, 1)
0.5
>>> clamp(3, -1, 1)
1
>>> clamp(-3, -1, 1)
-1
```

3. For each of the following while loops indicate whether the loop is guaranteed to terminate or not execute (indicate with “T”), is guaranteed to be an infinite loop (indicate with “I”) or depends (indicate with “D”). [4 points]

a. 
```
i = 1
while i != 0:
    i = i * -1
```

b. 
```
i = 5
while i > 0:
    i = i - 2
```

c. 
```
i = 3
while i < 5:
    i = int(input("Your age?"))
```

d. 
```
while 2 > 1:
    break
```
Numeric Operators
+,-, /, *: Addition, subtraction, division, multiplication
//: Floor division: Round division result down to nearest whole number
%: Modulus: Evaluate to remainder of division

• Range
\texttt{range(stop)}: Equivalent to \texttt{range(0, stop, 1)}
\texttt{range(start, stop[, step])}: Create sequence of integers from inclusive \texttt{start} to exclusive \texttt{stop} by \texttt{step}

• Slicing
\texttt{seq[start[:stop][:step]]}: Slice sequence \texttt{seq} from inclusive \texttt{start} to exclusive \texttt{stop} by \texttt{step}

Input/Output
• Reading input from the user
\texttt{input(message)}: Displays message to the user and returns what the user typed as a string

Strings
• The following functions are built-in
\texttt{len(string)}: Returns the number of characters in the string
\texttt{int(string)}, \texttt{float(string)}: Converts numeric string to int or float
\texttt{str(object)}: Converts object, e.g. int or float to a string
\texttt{sorted(string)}: Returns the characters of the string as a list in sorted order

• String object methods
\texttt{upper()}, \texttt{lower()}, \texttt{capitalize()}: Returns a new upper or lower-cased, or 1^{st} letter upper-cased string
\texttt{find(some_string)}: Returns the first index that \texttt{some_string} occurs at in the string or -1 if not found
\texttt{find(some_string, index)}: Same as above, but starts searching at index
\texttt{replace(old, new)}: Return a copy of the string with all occurrences of old substituted with new
\texttt{startswith(prefix)}: Returns \texttt{True} if the string starts with \texttt{prefix}, \texttt{False} otherwise
\texttt{endswith(suffix)}: Returns \texttt{True} if the string ends with \texttt{suffix}, \texttt{False} otherwise
\texttt{strip()}: Returns a copy of the string with only the leading and trailing whitespace removed
\texttt{split()}: Return a list of the words in the string using whitespace as the delimiter

• String operators
\texttt{string1 + string2}: Returns a new string that is the concatenation of \texttt{string1} and \texttt{string2}
\texttt{string * int}: Returns a new string that is \texttt{string} repeated \texttt{int} times
\texttt{substr in string}: Returns \texttt{True} if \texttt{substr} is a substring of \texttt{string}, \texttt{False} otherwise

Modules
• \texttt{random module}
\texttt{randint(a, b)}: Return a random integer \texttt{N} such that \(a \leq N \leq b\)
\texttt{uniform(a, b)}: Return a random floating point number \texttt{N} such that \(a \leq N \leq b\)

• \texttt{math module}
\texttt{sqrt(num)}: Return the square root of \texttt{num}