

Closed book, closed notes, log out of computer! Cheat sheet on reverse. Please write neatly!

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

<p>a. _____</p> <pre>x = 36 if x < 3: x = 3 else: x = 9 if x < 27: x = 27</pre>	<p>b. _____</p> <pre>x = 36 if x < 3: x = 3 elif x < 9: x = 9 elif x < 27: x = 27</pre>
---	--

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]

```
>>> 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]

<p>a. _____</p> <pre>i = 1 while i != 0: i = i * -1</pre>	<p>b. _____</p> <pre>i = 5 while i > 0: i = i - 2</pre>
<p>c. _____</p> <pre>i = 3 while i < 5: i = int(input("Your age?"))</pre>	<p>d. _____</p> <pre>while 2 > 1: break</pre>

CS 150 Fall 2022 – Quiz 3 “Cheat Sheet”

Numeric Operators

+, **-**, **/**, *****: Addition, subtraction, division, multiplication

//: Floor division: Round division result down to nearest whole number

%: Modulus: Evaluate to remainder of division

- Range

range(stop): Equivalent to `range(0, stop, 1)`

range(start, stop[, step]): Create sequence of integers from inclusive **start** to exclusive **stop** by **step**

- Slicing

seq[start[:stop[:step]]]: Slice sequence **seq** from inclusive **start** to exclusive **stop** by **step**

Input/Output

- Reading input from the user

input(message): Displays message to the user and returns what the user typed as a string

Strings

- The following functions are built-in

len(string): Returns the number of characters in the string

int(string), **float(string)**: Converts numeric string to int or float

str(object): Converts object, e.g. int or float to a string

sorted(string): Returns the characters of the string as a list in sorted order

- String object methods

upper(), **lower()**, **capitalize()**: Returns a new upper or lower-cased, or 1st letter upper-cased string

find(some_string): Returns the first index that **some_string** occurs at in the string or -1 if not found

find(some_string, index): Same as above, but starts searching at index

replace(old, new): Return a copy of the string with all occurrences of old substituted with new

startswith(prefix): Returns **True** if the string starts with prefix, **False** otherwise

endswith(suffix): Returns **True** if the string ends with suffix, **False** otherwise

strip(): Returns a copy of the string with only the leading and trailing whitespace removed

split(): Return a list of the words in the string using whitespace as the delimiter

- String operators

string1 + string2: Returns a new string that is the concatenation of string1 and string2

string * int: Returns a new string that is string repeated int times

substr in string: Returns **True** if substr is a substring of string, **False** otherwise

Modules

- **random** module

randint(a, b): Return a random integer N such that $a \leq N \leq b$

uniform(a, b): Return a random floating point number N such that $a \leq N \leq b$

- **math** module

sqrt(num): Return the square root of num