Chapter 10, 11, 12 Exercises.

Richard Sullivan Andison.

Chapter 10

9. In Python, indentation is used to indicate the exent of a block of code. What is the output of the following Python code?

```
# Creating variables first and second with values: 3,5
first = 3 second = 5
# if our variable first is smaller than second, print "second is bigger
if first < second:
        print ("second is bigger")
# If not, print "but this time
else:
        print ("but this time ")
# regardless of the previous if/else statements, print "first is bigger"
print("first is bigger")
```

The output of this code would be:
"Second is bigger
first is bigger"
since first (3) is less than second (5) The else statement does not run since first is indeed less than second. The final print statement "first is bigger" also runs since it is outside of the if/else statements and will run regardless.

There is also a newline (\n) between the two prints since by default, Python3 adds a newline to the end of each print statement.

Chapter 11

15. What are the different interpretations of the following English language sentence?

"I bought a shirt in the new store that was too large"

The sentence could be interpreted in these two ways:

1. **The shirt was bought at the new store and the shirt was too large.**
2. **The shirt was bought at the new store and the store itself is too large.**

Chapter 12

8. Find the output for the Turing machine

The first two instructions means that while the symbol is a 0 or 1, go to the next position to the right and change to state 2.
(1,1,1,2,R) - Change to state 2 and move 1 right
(1,0,0,2,R) - Change to state 2 and move 1 right

The only time a symbol is changed in state 1 is when there is a blank symbol, in which case, we overwrite the blank with a 1, change to state 2 and move one position right.
(1,b,1,2,R) - Change blank symbol to a 1 and change to state 2.

The last two instructions means that while in state 2, if the number is zero, move one space right and remain in state 2. If the symbol is 1, change to a zero and change to state 1.
(2,0,0,2,R) - Pass over and don't change state for all zeroes.
(2,1,0,1,R) - Change 1 to a 0, change to state one and move one position right.

when run on the tape:

...b1001b...

First, start at the leftmost non-blank number which is 1. Also noting we start in state 1. Our next instruction would be:
(1,1,1,2,R)

Which means:

if we are in state 1 and the current symbol is 1:
    Do not change the symbol
    Change to state 2
    Move one position to the right

...b1001b...

We are now at a 0 symbol in state 2 which means the next instruction would be:
(2,0,0,2,R)

Which means:

If we are in state 2 and the current symbol is a 0:
    Do not change the symbol
    Remain in state 2
    Move one position right

...b10<u>0</u>1b...

The previous step repeats since we are still in state 2 reading a 0

...b100<u>1</u>b...

We are now in state 2 with a symbol of 1 which means the next instruction is:
(2,1,0,1,R)

Which means:

If we are in state 2 and the current symbol is 1:
- Change the 1 to a zero
- Change to state 1
- Move one position right

...b1000<u>b</u>...

We are now in state 1 on a blank symbol which means our next instruction is:
(1,b,1,2,R)

Which means:

If we are in state 1 and the current symbol is blank:
- Change the blank symbol to a 1
- Change to state 2
- Move one position right

The resulting output would be **b10001**