1. Complete the table below that lays out syntax and use rules for the while keyword.

Rule	Correct Example	Incorrect Example
While loops must start with the while keyword	while x<5:	`loop x < 5:`
while loops must have a condition that can be iterated	`while count > 10:`	`while "nonsense":`
A colon must be included after the boolean expression	`while boolean_var:`	while boolean_var

All code considered part of the loop body must be indented one level deeper than the line with while	`while x < 10:`
	`  print("in loop")`
	`  x += 1`
		while x<10:
		print("in loop")
		x+=1

while loops shouldn't be an infinite loop	`while num > 100:`
	`  num -= 1`
	`print("done")`
		`while num > 100:`
		`  other_num -= 1`
		`  print("infinite!")`

Each loop body needs to include at least one line of code	while len(x) <= 4: print(x)	`while len(x) <= 4:` `#syntax error!`

2. Consider a scenario where you are writing a program that requires a yes/no answer from the user. Users are familiar with programs that allow them to use y, Y, yes, YES, etc. as possible ways to express a "yes" answer, and the same pattern for "no" answers. Write a pseudocode solution for this issue that reads a user's input, and determines whether their answer is something like yes or no. Account for any differences in capitalization (Python has a function that can turn everything lowercase), and continue asking the user for their answer until they provide valid input. Your pseudocode should use the idea of the while loop you learned about this week.

```
While True:
  input = input().lower()
  if input in ["no", "yes", "y", "n"]:
      break
  else:
      print("invalid")
print(input)
```

3. The code below is intended to print only even numbers up to 10, but currently does not complete this properly. Find and fix the bug(s) to achieve this goal and explain the problem(s) with the original program.

```
1 i = 0
2 while i < 10:
3     i + 1
4     if i % 2 == 1:
5         break
6     print(i)
```

When the if statement gets hit, it breaks then continues to print i since i is not in line with the break. The if statement should have less indentation.

4. Examine the code below

```
1 counter = 1
2 total = 0
3 while counter <= 6:
4     total = total + counter
5     counter = counter + 2
6 print(total)
```

1 4 9
3 5

(a) What does this program print?

A. 12 B. 9 C. 7 D. 8

(b) Explain what the while loop is doing here.

This loop is taking two variables, counter and total and iterating on it. Counter starts at 1, so the loop iterates this to equal 3 and eventually 5. total is iterated by combining itself and counter, making it equal 1, 4, and 9.

5. A programmer is trying to write a program that helps build a student's class schedule, by asking them about the classes they are taking. Their code so far looks like this:

```
1 classDept = input("Please enter the course department: (e.g. CSCI)")
2 classNum = int(input("Please enter the course number: (e.g. 128)"))
3 classSection = input("Please enter the course section: (e.g. A)")
```

However, they don't know how many classes the student is going to take, and can assume that different students will be taking different numbers of classes. (And probably more than one). Describe in your own words how the programmer might rewrite their program to account for this. The programmer can ask for an initial input for # of classes, then have a loop that uses the input to constantly check if all classes were registered.

6. Consider the following problem:

Given a positive number n and a digit d, insert d into n at the location that makes the resulting number as large as possible (including the ends). For example, with $n = 2034$ and $d = 1$, we could make $\mathbf{1}2034, 2\mathbf{1}034, 20\mathbf{1}34, 203\mathbf{1}4, 2034\mathbf{1}$. The largest of these is 21034, so that is the answer.

n can have up to 10^9 **digits**, so trying every possible solution (as we did in the previous example) will be inefficient and slow. Instead, create some examples and see if you can spot a pattern. Use this pattern to devise a better algorithm to solve this problem.

(a) After determining a better solution than brute force, write 1-2 sentences describing the key idea if you were going to solve this problem using your new solution *by hand*. This should **not** include any references to data structures, implementation, or Python.

(b) Once you are certain your idea is correct, implement it. Write pseudocode for your solution below. Hint: do not convert n to an integer. Work with the string representation instead.