1. Complete the code to output the statement, "Diego's favorite food is lasagna". Remember that precise syntax must be used to receive credit. **1 point**

```
name = "Diego"
fav_food = "lasagna"
print(name + "'s favorite food is " + fav_food)

```

Run
Reset

```
Diego's favorite food is lasagna
```

2. What's the value of this Python expression: `7 < "number"`? **1 point**

- ○ True
- ○ False
- ◉ TypeError
- ○ 0

3. What directly follows the elif keyword in an elif statement? **1 point**

- ○ A function definition
- ◉ A comparison
- ○ A colon
- ○ A logical operator

4. Consider the following scenario about using if-elif-else statements: **1 point**

Police patrol a specific stretch of dangerous highway and are very particular about speed limits. The speed limit is 65 miles per hour. Cars going 80 miles per hour or more are given a "Reckless Driving" ticket. Cars going more than 65 miles per hour are given a "Speeding" ticket. Any cars going less than that are labeled "Safe" in the system.

Fill in the blanks in this function so it returns the proper ticket type or label.

```
def speeding_ticket(speed):
    if speed > 80:
        ticket = "Reckless Driving"
    elif speed > 65:
        ticket = "Speeding"
    else:
        ticket = "Safe"
    return ticket


print(speeding_ticket(87)) # Should print Reckless Driving
print(speeding_ticket(66)) # Should print Speeding
print(speeding_ticket(65)) # Should print Safe
print(speeding_ticket(85)) # Should print Reckless Driving
print(speeding_ticket(35)) # Should print Safe
print(speeding_ticket(77)) # Should print Speeding
```

Run
Reset

```
Reckless Driving
Speeding
Safe
Reckless Driving
Safe
Speeding
```

5. What's the value of the comparison in this **if** statement? Hint: The answer is not what the code will print. **1 point**

```
n = 4
if n*6 > n**2 or n%2 == 0:
    print("Check")

```

- ○ True
- ○ False
- ◉ Check
- ○ 24 > 16 or 0

6. Fill in the blanks to complete the function. The "identify_IP" function receives an "IP_address" as a string through the function's parameters, then it should print a description of the IP address. Currently, the function should only support three IP addresses and return "unknown" for all other IPs. **1 point**

```
def identify_IP(IP_address):
    if IP_address == "192.168.1.1":
        IP_description = "Network router"
    elif IP_address == "8.8.8.8" or IP_address == "8.8.4.4":
        IP_description = "Google DNS server"
    elif IP_address == "142.250.191.46":
        IP_description = "Google.com"
    else:
        IP_description = "unknown"
    return IP_description


print(identify_IP("8.8.4.4")) # Should print 'Google DNS server'
print(identify_IP("142.250.191.46")) # Should print 'Google.com'
print(identify_IP("192.168.1.1")) # Should print 'Network router'
print(identify_IP("8.8.8.8")) # Should print 'Google DNS server'
print(identify_IP("10.10.10.10")) # Should print 'unknown'
print(identify_IP("")) # Should Should print 'unknown'
```

Run

Reset

```
Google DNS server
Google.com
Network router
Google DNS server
unknown
unknown
```

7. Can you calculate the output of this code? **1 point**

```
def difference(x, y):
    z = x - y
    return z


print(difference(5, 3))
```

2

8. What's the value of this Python expression? **1 point**

```
((24 == 5*2) and (24 > 3*5) and (2*6 == 12))
```

- ○ True
- ◉ False
- ○ 15
- ○ 10

9. Fill in the blanks to complete the function. The "make_positive" function takes in a number and converts that number to its positive equivalent. Complete the function to accomplish the following tasks: **1 point**

- use an if statement to test if the number is negative;
- use a calculation inside the if statement to change the negative number to be positive;
- use a calculation in the else statement to return any positive "number" unchanged.

```
def make_positive(number):
    if number < 0:
        result = number * -1
    else:
        result = number
    return result


print(make_positive(-4))   # Should print 4
print(make_positive(0))    # Should print 0
print(make_positive(-.25)) # Should print 0.25
print(make_positive(5))    # Should print 5
```

Run

Reset

```
4
0
0.25
5
```

10. Which of the following are good coding-style habits? Select all that apply. **1 point**

- ☐ Writing code using the least amount of characters as possible
- ☑ Adding comments
- ☑ Cleaning up duplicate code by creating a function that can be reused
- ☑ Refactoring the code

Upgrade to submit