Reading and Writing

1) Give an example of when we used **digitalRead()**.

2) **digitalRead()** returns either _____ or _____ .

3) Give an example of when we used **analogRead()**.

4) **analogRead()** returns a value between _____ and _____ .

5) Give an example of when we used **digitalWrite()**.

6) **digitalWrite()** sets the voltage to either _____ or _____ .

7) Give an example of when we used **analogWrite()**.

8) **analogWrite()** sets a voltage between _____ and _____ .

9) **analogWrite()** can be used with the following pins:

Arguments

1) Fill out the function, **longShort()**, that takes a single argument- delay time for first blink. The second blink should delay *half as long* as the first.

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void draw() {
  longShort(500);
  longShort(800);
}

void (int delayTime) {
  // first blink
  digitalWrite(ledPin, HIGH);

  delay(                        );

  digitalWrite(ledPin, LOW);

  delay(                        );


  // second blink
  digitalWrite(ledPin, HIGH);

  delay(                        );

  digitalWrite(ledPin, LOW);

  delay(                        );

}
```

Return

1) The function **returnSum()** *returns* a value whereas **printSum()** just prints the sum. When returning values, instead of declaring "void," we specify the type of variable the function is returning.

What does *Serial.println(s0 + s1)* equal? _____

```
void setup() {
  Serial.begin(9600);
}

void draw() {
  int s0 = returnSum(1, 3);
  int s1 = returnSum(2, 4);
  Serial.println(s0 + s1);
}

int sum(int a, int b) {
  return a + b;
}
```

2) Write a function, **square()**, that takes a single argument and returns the square of that number.

```
void setup() {
  Serial.begin(9600);
}

void draw() {
  Serial.println(square(4));
  Serial.println(square(5));
}

// square function goes here




}
```

Conditionals

1) Check if a number is equal to your favorite number. If it is, turn on the LED. If it isn't, turn off the LED.

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void draw() {
  // is 13 your favorite number?
  checkFavoriteNumber(13);
}

void checkFavoriteNumber(int guess) {

  if (                              ) {



  }
  else {



  }
}
```

2) Fill out the following function, **checkPiezoHit()** so that if:

- hit is less than 100, turn LED off.
- hit is less than 200, turn LED 10% on.
- hit is less than 500, turn LED 50% on.
- hit is over 500, turn LED 100% on.

```
int ledPin = 11;
int piezoPin = A0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(piezoPin, INPUT);
}

void draw() {
  checkPiezoHit();
}

void checkPiezoHit() {
  int hit = analogRead(piezoPin);

  if (            ) {

    analogWrite(ledPin, 0);
  }
  else if (          ) {



  }
  else if (          ) {



  }
  else {



  }
}
```

3) Check a light sensor. If its value is less than 400, blink an LED. Otherwise, turn it off.

```
int ledPin = 13;
int lightPin = A0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void draw() {
  checkLight();
}

void checkLight() {
  int lightLevel = analogRead(lightPin);




}
```

Loops

1) Use a for loop to stay on for an increasing number of seconds: 1 second, 2 seconds, 3 seconds, ... , 10 seconds.

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  delayIncrease();
}

void delayIncrease() {

  // fill this out

  for (int i =      ; i              ; i++) {

    digitalWrite(ledPin, HIGH);

    // fill this out

    delay(          );

    digitalWrite(ledPin, LOW);
    delay(1000);
  }
}
```

2) Write a function **printRange()** that takes two arguments- start and end- and uses a for loop to print out all of the numbers between the start and end.

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  printRange(3, 6);   // prints: 3, 4, 5
  printRange(1, 5);   // prints: 1, 2, 3, 4
}

void printRange(int start, int end) {



}
```