Assignment 1

07/01/2025

1a

You need 4 bits.

It's impossible because a lot of the combinations already are used for letters.

1b

a=0

Loop 0: a=2, Loop 1: a=4, Loop 2: a=8, Loop 3: a=16, Loop 4: a=32

1c

```
const uint8_t ledPin = 12; // defining the ledPin
const int tU = 200; //timeUnit


void led(uint8_t Type) {  // High 1 or Low 0
  if (Type == 1) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}

void DD(uint8_t len) {  //Dot or Dash
  if (len == 0){ // dot
    led(1);
    delay(tU);
    led(0);
  }
  else {   // dash
    led(1);
    delay(tU*3);
    led(0);
  }
  delay(tU);
}
```

```
int S[] = {0, 0, 0};
int O[] = {1, 1, 1};



void morse(int letter[], uint8_t len) {

  for (int i = 0; i < len; i++){
    DD(letter[i]);
  }
}

void setup() {

  pinMode(ledPin, OUTPUT); // initialize digital pin LED_BUILTIN as an output.
}

void loop() {

  morse(S, sizeof(S) / sizeof(S[0]));
  morse(O, sizeof(O) / sizeof(O[0]));
  morse(S, sizeof(S) / sizeof(S[0]));

}
```

1d

It uses the built in LED

1e

```
int A[] = {0, 1};
int L[] = {0, 1, 0, 0};
int I[] = {0, 0};


void morse(int letter[], uint8_t len) {

  for (int i = 0; i < len; i++){
    DD(letter[i]);
  }
}

void setup() {

  pinMode(ledPin, OUTPUT); // initialize digital pin LED_BUILTIN as an output.
}

void loop() {

  morse(A, sizeof(A) / sizeof(A[0]));
  morse(L, sizeof(L) / sizeof(L[0]));
  morse(I, sizeof(I) / sizeof(I[0]));

}
```

1f

I guess I already did?

2a

42 % 5 = 2

2b

```
uint8_t green = 7;
uint8_t yellow = 8;
uint8_t red = 12;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  pinMode(green, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(red, OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(red, HIGH);
  Serial.println("STOOOP!");
  delay(1000);
  digitalWrite(red, LOW);

  digitalWrite(yellow, HIGH);
  Serial.println("waaaaaiiiit");
  delay(1000);
  digitalWrite(yellow, LOW);

  digitalWrite(green, HIGH);
  Serial.println("GO! GO! GO!");
  delay(1000);
  digitalWrite(green, LOW);

}
```

2c

```
uint8_t green = 7;
uint8_t yellow = 8;
uint8_t red = 12;
const int o = HIGH;
const int z  = LOW;


void OnOff(char One, char Two, char Three) {
  digitalWrite(green, One);
  digitalWrite(yellow, Two);
  digitalWrite(red, Three);
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  pinMode(green, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(red, OUTPUT);
}
```

```
void loop() {
  OnOff(z,z,z);
  Serial.println("0");
  delay(500);

  OnOff(z, z, o);
  Serial.println("1");
  delay(500);

  OnOff(z, o, z);
  Serial.println("2");
  delay(500);

  OnOff(z, o, o);
  Serial.println("3");
  delay(500);

  OnOff(o, z, z);
  Serial.println("4");
  delay(500);

  OnOff(o, z, o);
  Serial.println("5");
  delay(500);

  OnOff(o, o, z);
  Serial.println("6");
  delay(500);

  OnOff(o, o, o);
  Serial.println("7");
  delay(500);
```

2d

You could probably make some counter, and have them change when %2, %3, %4 = 0 with respect to led 1, 2, 3.

3a

Input is HIGH when HIGH

Input_pullup is HIGH when LOW

3b

inversion, for example != means not equal.

3c

```
uint8_t green = 7;
uint8_t yellow = 8;
uint8_t red = 12;
uint8_t but = 13;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  pinMode(green, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(but, INPUT_PULLUP);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (digitalRead(but) == HIGH){
    digitalWrite(green, HIGH);
  }
  else {
    digitalWrite(green, LOW);
  }
}
```

```
uint8_t green = 7;
uint8_t yellow = 8;
uint8_t red = 12;
uint8_t but = 13;

bool lastButtonState = LOW; // Tracks the previous button state
bool ledState = LOW;        // Tracks the current LED state

void setup() {
  Serial.begin(115200);
  pinMode(green, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(but, INPUT_PULLUP);
}

void loop() {
  bool currentButtonState = digitalRead(but);

  if (currentButtonState == LOW && lastButtonState == HIGH) {

    ledState = !ledState;
    digitalWrite(green, ledState);
    delay(50);
  }

  // Update the button state
  lastButtonState = currentButtonState;
}

```

3d

It checks every 50ms, seems fair.

4