

**DO NOT WRITE ON THIS PAGE. FILL IN THE MISSING CODE ON THE ANSWER SHEET.**

**Course 4**  
**Stage 18: Artist Binary**  
Fill in the missing code.

**Stage 18: Artist Binary**

**Puzzle 1**

```
var binary;

var bit;

var i;

function draw(binary) {

  if (!bit) {

    bit = 0;

  }

  var i_end = binary.length;

  var i_inc = 1;

  if (1 > i_end) {

    i_inc = -i_inc;

  }

  for (i = 1;

    i_inc >= 0 ? i <= i_end : i >= i_end;

    i += i_inc) {

    if (binary.charAt(i - 1) == 1) {

      draw_bit();

    }

  }

}
```

```

    jumpForward(50);

    bit = bit + 1;

    if (bit > 7) {

        jumpBackward(400);

        turnRight(90);

        jumpForward(50);

        turnLeft(90);

        bit = 0;

    }

}

}

function draw_bit() {

    penWidth(25);

    for (var count2 = 0; count2 < 4; count2++) {

        moveForward(25);

        turnRight(90);

    }

}

for (var count = 0; _____; count++) {

    draw('10101010');

    draw('01010101');

}

```

## Puzzle 2

```
var binary;

var bit;

var i;

function draw(binary) {

  if (!bit) {

    bit = 0;

  }

  var i_end = binary.length;

  var i_inc = 1;

  if (1 > i_end) {

    i_inc = -i_inc;

  }

  for (i = 1;

    i_inc >= 0 ? i <= i_end : i >= i_end;

    i += i_inc) {

    if (binary.charAt(i - 1) == 1) {

      draw_bit();

    }

    jumpForward(50);

    bit = bit + 1;

    if (bit > 7) {

      jumpBackward(400);

      turnRight(90);

      jumpForward(50);

    }

  }

}
```

```

    turnLeft(90);

    bit = 0;
  }
}

function draw_bit() {
  penWidth(25);
  for (var count2 = 0; count2 < 4; count2++) {
    moveForward(25);
    turnRight(90);
  }
}

for (var count = 0; count < 8; count++) {
  draw('_____');
}

```

### Puzzle 3

```

var binary;
var bit;
var i;

function draw(binary) {
  if (!bit) {
    bit = 0;

```

```
}  
  
var i_end = binary.length;  
  
var i_inc = 1;  
  
if (1 > i_end) {  
    i_inc = -i_inc;  
}  
  
for (i = 1;  
    i_inc >= 0 ? i <= i_end : i >= i_end;  
    i += i_inc) {  
    if (binary.charAt(i - 1) == 1) {  
        draw_bit();  
    }  
    jumpForward(50);  
    bit = bit + 1;  
    if (bit > 7) {  
        jumpBackward(400);  
        turnRight(90);  
        jumpForward(50);  
        turnLeft(90);  
        bit = 0;  
    }  
}  
}  
  
function draw_bit() {  
    penWidth(25);
```

```

for (var count2 = 0; count2 < 4; count2++) {
    moveForward(25);
    turnRight(90);
}
}

for (var count = 0; count < 21; count++) {
    draw('_____');
}

```

## Puzzle 4

```

var binary;
var bit;
var i;

function draw(binary) {
    if (!bit) {
        bit = 0;
    }

    var i_end = binary.length;
    var i_inc = 1;
    if (1 > i_end) {
        i_inc = -i_inc;
    }

    for (i = 1;
        i_inc >= 0 ? i <= i_end : i >= i_end;

```

```
    i += i_inc) {  
    if (binary.charAt(i - 1) == 1) {  
        draw_bit();  
    }  
    jumpForward(50);  
    bit = bit + 1;  
    if (bit > 7) {  
        jumpBackward(400);  
        turnRight(90);  
        jumpForward(50);  
        turnLeft(90);  
        bit = 0;  
    }  
    }  
}  
  
function draw_bit() {  
    penWidth(25);  
    for (var count = 0; count < 4; count++) {  
        moveForward(25);  
        turnRight(90);  
    }  
}  
  
draw('00000000');  
draw('01100110');
```

```
draw('01100110');  
  
draw('00000000');  
  
draw('_____');  
  
draw('01000010');  
  
draw('00111100');  
  
draw('00000000');
```

## Puzzle 5

```
var binary;  
  
var bit;  
  
var i;  
  
function draw(binary) {  
  if (!bit) {  
    bit = 0;  
  }  
  
  var i_end = binary.length;  
  
  var i_inc = 1;  
  
  if (1 > i_end) {  
    i_inc = -i_inc;  
  }  
  
  for (i = 1;  
    i_inc >= 0 ? i <= i_end : i >= i_end;  
    i += i_inc) {  
    if (binary.charAt(i - 1) == 1) {  
      draw_bit();  
    }  
  }  
}
```



```
}

jumpForward(50);

bit = bit + 1;

if (bit > 7) {

    jumpBackward(400);

    turnRight(90);

    jumpForward(50);

    turnLeft(90);

    bit = 0;

}

}

}

function draw_bit() {

    penWidth(25);

    for (var count2 = 0; count2 < 4; count2++) {

        moveForward(25);

        turnRight(90);

    }

}

for (var count = 0; count < 11; count++) {

    draw('110010110');

    draw('_____');

}

}
```

## Puzzle 6

```
var binary;

var bit;

var i;

function draw(binary) {

  if (!bit) {

    bit = 0;

  }

  var i_end = binary.length;

  var i_inc = 1;

  if (1 > i_end) {

    i_inc = -i_inc;

  }

  for (i = 1;

    i_inc >= 0 ? i <= i_end : i >= i_end;

    i += i_inc) {

    if (binary.charAt(i - 1) == 1) {

      draw_bit();

    }

    jumpForward(50);

    bit = bit + 1;

    if (bit > 7) {

      jumpBackward(400);

      turnRight(90);

      jumpForward(50);

    }

  }

}
```

```

    turnLeft(90);

    bit = 0;

}

}

}

function draw_bit() {

    penWidth(25);

    for (var count2 = 0; count2 < 4; count2++) {

        moveForward(25);

        turnRight(90);

    }

}

for (var count = 0; count < 13; count++) {

    draw('_____');

}

```

## Puzzle 7

```

var binary;

var bit;

var i;

function draw_bit() {

    penWidth(25);

    for (var count = 0; count < 4; count++) {

```

```
    moveForward(25);

    turnRight(90);
}
}

function draw(binary) {

    if (!bit) {

        bit = 0;

    }

    var i_end = binary.length;

    var i_inc = 1;

    if (1 > i_end) {

        i_inc = -i_inc;

    }

    for (i = 1;

        i_inc >= 0 ? i <= i_end : i >= i_end;

        i += i_inc) {

        if (binary.charAt(i - 1) == 1) {

            draw_bit();

        }

        jumpForward(50);

        bit = bit + 1;

        if (bit > 7) {

            jumpBackward(400);

            turnRight(90);

            jumpForward(50);
```

```
    turnLeft(90);

    bit = 0;

  }

}

}

bit = 0;

draw('0000001');

draw('01111101');

draw('01000101');

draw('01010101');

draw('01010101');

draw('01011101');

draw('0100001');
```

**Puzzle 8 (write down the URL of your finished design on the answer sheet)**

**<http://studio.code.org/c/>\_\_\_\_\_**

**Puzzle 9 (write down the URL of your finished design on the answer sheet)**

**<http://studio.code.org/c/>\_\_\_\_\_**