

Directions: Answer each of the following questions in complete sentences and/or code.

1. Under what conditions does the code within a while loop execute?

only when Boolean condition is TRUE

2. What will be printed to the console after the following code segment has executed?

```
var x = 0;
while( x < 5)
{
    x++;
    console.log(x);
}
console.log("The final value of x is: " + x );
```

X = 1 2 3 4 5

OUTPUT

*1
2
3
4
5*

The final value of x is 5

3. Write an example of a while loop that will loop infinitely.

```
var x = 0;
while (x > -1)
{ //code
    x++; // #4: change this to x--;
}
```

4. Fix the while loop you wrote in the previous problem
(use comments // to show what you would change).

5. What will be printed to the console after the following code segment has executed?

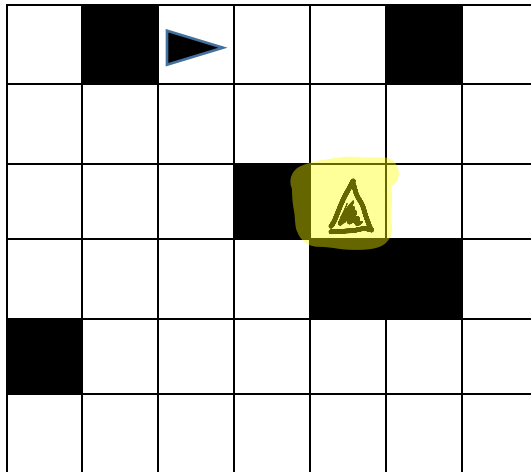
```
var value = 20;
while(value != 5)
{
    if(//value is even)
        value = value / 2;
    else
        value = value++;
}
console.log(value);
```

value = 20 10 5

OUTPUT
5

6. A robot is represented as a triangle in a grid of squares. The initial position and direction of the robot is shown below. The robot can only move onto a white square (not a black square). Show the final position and direction of the robot on the grid after completing the following code segment.

```
REPEAT 3 TIMES
{
    REPEAT UNTIL (NOT CAN_MOVE(forward))
    {
        MOVE _FORWARD();
    }
    ROTATE_RIGHT();
}
```



final position
highlighted.
facing ↑

7. Consider the pseudocode below. If the intention of the code is to see how many flips of a coin it takes get 10 heads in a row, what should replace //missing code?

```
headsCount ← 0
numFlips ← 0
WHILE ( headsCount < 10 )
{
    flip ← randomNumber(0,1)  //0 = tails, 1 = heads

    //missing code

    IF (current flip is a heads)
        headsCount ← headsCount + 1
    ELSE
        headsCount ← 0
}
DISPLAY (numFlips)
```

numFlips++; (we must increase the
of flips by one each
time)

8. What will be printed as a result of executing the following code?
(The original array is shown)

index:	1	2	3	4	5
nums:	2	6	3	8	9

```
nums = [2, 6, 3, 8, 9];  
var x = nums[3];  
var y = nums[1];  
console.log(x + y);
```

// x = 8
// y = 6

5

9. Using JavaScript, complete the following function `allSums` which returns the sum of all values in the parameter array `points`. Reminder, in JavaScript, index values begin at 0.

```
function allSums( points )  
{
```

```
    var sum = 0;
```

```
    //complete the method
```

```
    for (var i = 0; i < points.length; i++)  
    { sum = sum + points[i]; }
```

```
    return sum;
```

```
}
```

10. State whether each of the following for loop headers would cause an out of bounds error in JavaScript for an array called `ray`? (Write YES for bounds error, and NO if no error will occur)

a) for (var i = 0; i < ray.length; i++)

NO

b) for (var i = 0; i <= ray.length; i++)

YES

c) for (var i = 0; i <= ray.length - 1; i++)

NO

d) for (var i = ray.length; i >= 0; i--)

YES

e) for (var i = ray.length - 1; i >= 0; i--)

NO

code that causes errors are highlighted

11. What is store in the list `vals` after the following code segment is run?

index:	1	2	3
vals:	2	5	7


```

vals <- [2,5,7]
APPEND(vals, 12)
INSERT(vals, 1, 8)
INSERT(vals, 1, 9)
APPEND(vals, 4)
REMOVE(vals, 2)
INSERT(vals, 1, 10)
INSERT(vals, 4, 6)
REMOVE(vals, 3)
DISPLAY(vals)

```

vals

2, 5, 7
 2, 5, 7, 12
 8, 2, 5, 7, 12
 9, 8, 2, 5, 7, 12
 9, 8, 2, 5, 7, 12, 4
 9, 2, 5, 7, 12, 4
 10, 9, 2, 5, 7, 12, 4
 10, 9, 2, 6, 5, 7, 12, 4
 10, 9, 6, 5, 7, 12, 4
 → [10, 9, 6, 5, 7, 12, 4]

12. Consider the algorithm to swap two elements (position A and position B) in an array.

STEP 1: Store the value from position A into a temporary variable

STEP 2: Assign the value from position B to position A

STEP 3: Assign the value from the temporary variable to position B

Use the algorithm to write the function (in JavaScript) called `swap` (shown below). Reminder: in JavaScript, arrays start with index 0. A call to the function and the result is shown:

```

var data = [10, 41, 26, 18, 39];
swap(data, 2, 4);      //results in data now containing: [10, 41, 39, 18, 26]

```

```

function swap(anArray, posA, posB)
{

```

```

    var temp = anArray[posA];
    anArray[posA] = anArray[posB];
    anArray[posB] = temp;
}

```

13. What is the purpose of the PROCEDURE below?

```
PROCEDURE mystery (data, target)
{
  count = 0
  i = 1
  REPEAT UNTIL (i = LENGTH(data))
  {
    IF (data[i] > target)
    {
      count = count + 1
    }
    i = i+1
  }
  RETURN (count)
}
```

return the # of
values in data
array that are
greater than
target

14. An array called odds is created and initialized. What does odds store after the following JavaScript code executes?

```
var i = 0; 0 1 2 3 4 5 6 7 length=8
var odds = [3, 9, 7, 11, 15, 17, 11, 5];
while (i < odds.length - 1)
{
  if(odds[i] < odds[i+1])
  {
    odds[i] = odds[i + 1];
    i = i + 2;
  }
  i++;
}
```

i	odds[i]	odds[i+1]	odds
0	3	9	9, 9, 7, 11, 15, 17, 11, 5
2	7	11	9, 9, 11, 11, 15, 17, 11, 5
4	15	17	9, 9, 11, 11, 17, 17, 11, 5
6	11	5	
7			

odds: [9, 9, 11, 11, 17, 17, 11, 5]

15. Why would we use computer simulations to model real world events?

Real world events can take too long, cost too much money, or be dangerous to humans.

