

1. Given the code snippet, answer the questions that follow:

```

1 let tar = (n: number): number => {
2   return n ** 2 / 9;
3 };
4
5 let heel = (n: number): number => {
6   return n * 4 / 3;
7 };
8
9 let born = (n: number, m: number): number => {
10  if (n % 4 === 0) {
11    return n;
12  } else {
13    return m;
14  }
15 };
16
17 let main = async () => {
18   let values: number[] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15];
19   let groupA: number[] = [];
20   let groupB: number[] = [];
21   let groupC: number[] = [];
22   for (let i = 0; i < values.length; i++) {
23     if (values[i] % 3 === 0) {
24       groupA[groupA.length] = tar(values[i]);
25     }
26     if (values[i] % 6 === 0) {
27       groupB[groupB.length] = heel(values[i]);
28     } else if (values[i] % 2 === 0) {
29       groupC[groupC.length] = born(values[i], values[values[i]]);
30     }
31   }
32 };
33
34 main();

```

1.1 After running the code snippet above, what are the values held inside **groupA**? Fill out the value of each index below. Leave empty indices blank.

1	4	9	16	25	
0	1	2	3	4	5

1.2 After running the code snippet to the left, what are the values held inside **groupB** at each index. Leave empty indices blank.

8	16				
0	1	2	3	4	5

1.3 After running the code snippet to the left, what are the values held inside **groupC** at each index. Leave empty indices blank.

3	4	8	11	15	
0	1	2	3	4	5

1.4 How many times is the born function called?

**5**

2. Answer the questions according to the code on the left.

```
1 let main = async () => {
2   let arr = [hey(3), wassup(10), hello(4)];
3   print(arr);
4 };
5
6 let hey = (x: number): number => {
7   print("0");
8   let y = wassup(4);
9   return x + y;
10 };
11
12 let wassup = (x: number): number => {
13   print("1");
14   x = x + 7;
15   return x;
16 };
17
18 let hello = (x: number): number => {
19   let y = hey(3);
20   wassup(8);
21   print("1");
22   print("0");
23   return x * y;
24 };
```

2.1 What is the value of `arr` after the main function executes?

14	17	56
0	1	2

2.2 What is the printed output after the main function executes?

**0,1,1,0,1,1,1,0**

3. Write a function called `oddAndEven` that takes in a number array `arr` and returns a number. This function should 1) print elements of the array that are odd and have an index that is divisible by 2 and 2) return the number of elements printed. To receive credit for this question you must use a loop.

```
let oddAndEven = (arr: number[]): number => {
  let i = 0;
  let count = 0;
  while (i < arr.length) {
    if ((arr[i] % 2 !== 0) && (i % 2 === 0)) {
      print(arr[i]);
      count = count + 1;
    }
    i++;
  }
  return count;
}
```

3.1 If `arr = [2, 9, 4, 17, 9, 10, 15, 13, 14, 21]`, what is the printed output of `oddAndEven(arr)`?

**9, 15**

3.2 What is the return value?

**2**

4. Given the following function `foo`, answer the questions that follow.

```
1 let foo = (n: number): number => {
2   let x = 0;
3   while (n > 0) {
4     let r = n % 10;
5     x = x + r;
6     r = r / 10;
7     n = n / 10 - r;
8   }
9   return x;
10 }
```

4.1 What is the output of calling `print(foo(12));`?

**3**

4.2 What is the output of calling `print(foo(342));`?

**9**

4.3 What is the output of calling `print(foo(327));`?

**12**

4.4 In one sentence, what does the `foo` function do? **Sums the digits of a number.**

5. Given the following code, answer the questions that follow. Hint: strings can be treated like arrays. For example, `"string"[0]` evaluates to `"s"`, and `"string".length` evaluates to `6`.

```
1 let arraySmasher = (foo: string): string => {
2   let s = "";
3   for (let b = 0; b < foo.length; b++) {
4     if (s.length % 2 === 0) {
5       s = s + foo[b] + "oo";
6     } else {
7       s = s + foo[b] + "eo";
8     }
9   }
10  return s;
11 };
```

5.1 What is the value of `arraySmasher("hiya")`?  
**hooieoyooaao**

5.2 Write a function call to `arraySmasher` that would return the value `"ooooeo"`.

**arraySmasher("oo");**

5.3 Write a function call to `arraySmasher` that would return the value `"booyeoeeo"`.

**arraySmasher("bye");**

6. There are many formal mathematical sequences that have fancy names and applications, and in this question you will write a loop that computes the Collatz sequence.

1. Write a function named `collatz` that takes in a number called `n`.
2. The function should return nothing.
3. The function should have a for-loop that loops as long as the value of `n` is greater than 1.
4. Within each iteration of the loop:
  - (a) If `n` is even, reassign `n` to be itself divided by two.
  - (b) Otherwise, reassign `n` to be  $3 * n + 1$ .
  - (c) Print the value of `n`.

```
let collatz = (): void => {
  let n = 12;
  for (let i = 0; n > 1; i++) {
    if (n % 2 === 0) {
      n = n / 2;
    } else {
      n = 3 * n + 1;
    }
    print(n);
  }
};
```