

Recursion in Pseudo-code

Question 1

The base case for getting the sum of a list of integers is when $n = 0$.

Therefore we check to see if n is 0 and return 0 if this is the case.

Otherwise we return the sum of the last number plus the sum of all the previous numbers.

e.g

```
// Recursive function
sumRecursive(num) {
```

```
    if (num == 0)
        return 0
```

```
    lastNumber = num
    allButLastSum = sumRecursive(num - 1)
```

```
    return lastNumber + allButLastSum
```

```
}
```

```
// Input
number = 5
```

```
sumRecursive(5)
```

```
// Output
15
```

Question 2

e.g

```
// Recursive function
numStepsRecursive(distance) {

    // Base case. If step is less than 1m we instead take a 1m step
    if (distance < 1)
        return 1

    // Pass in half the distance each time to the recursive function
    stepsCount = numStepsRecursive(distance / 2)

    // Add 1 since the split in half rule will never get us all the way to the destination and
    // thus we need to take one more step
    return stepsCount + 1

}
```

```
// Input
distance = 100 // Distance in metres
```

```
sumRecursive(distance)
```

```
// Output
8
```

```
// Proof of steps count with distance = 100
```

Step 1:	distance/2 = 50m	distance travelled: 50m
Step 2:	distance/2 = 25m	distance travelled: 75m
Step 3:	distance/2 = 12.5m	distance travelled: 87.5m
Step 4:	distance/2 = 6.25m	distance travelled: 93.5m
Step 5:	distance/2 = 3.125m	distance travelled: 96.875m
Step 6:	distance/2 = 1.5625m	distance travelled: 98.4375m
Step 7:	(distance/2 < 1) => 1m	distance travelled: 99.4375m
Step 8:	(distance/2 < 1) => 1m	distance travelled: 100.4375