

Week 12 — Recursion

Anh Huynh

1. The following function prints every other number from a low number to a high number. For example, if low is 0 and high is 10, it would print:

```
1 0
2 2
3 4
4 6
5 8
6 10
```

Identify the base case in the function:

```
1 def print_every_other(low, high)
2   return if low > high
3   puts low
4   print_every_other(low + 2, high)
5 end
```

Base case: return if low > high; this stops the recursion once low passes high

2. My kid was playing with my computer and changed my factorial function so that it computes factorial based on (n - 2) instead of (n - 1). Predict what will happen when we run factorial(10) using this function:

```
1 def factorial(n)
2   return 1 if n == 1
3   return n * factorial(n - 2)
4 end
```

```
factorial(10)
= 10 * factorial(8)
= 10 * 8 * factorial(6)
= 10 * 8 * 6 * factorial(4)
= 10 * 8 * 6 * 4 * factorial(2)
= 10 * 8 * 6 * 4 * 2 * factorial(0)
```

Since the kid changed (n-1) to (n-2), the function never reaches the base case n == 1. Instead, it eventually reaches n == 0 but there is no base case for 0. Because 0 is not handled, the function keeps calling itself with negative numbers, causing infinite recursion and eventually stack overflow.

3. Following is a function in which we pass in two numbers called low and high. The function returns the sum of all

the numbers from low to high. For example, if low is 1, and high is 10, the function will return the sum of all numbers from 1 to 10, which is 55. However, our code is missing the base case, and will run indefinitely! Fix the code by adding the correct base case:

```
1 def sum(low, high)
2     return high + sum(low, high - 1)
3 end
```

fix:

```
1 def sum(low, high)
2     return 0 if high < low
3     return high + sum(low, high - 1)
4 end
```

4. Here is an array containing both numbers as well as other arrays, which in turn contain numbers and arrays:

```
1 array=[ 1,
2         2,
3         3,
4         [4, 5, 6],
5         7,
6         [8,
7           [9, 10, 11,
8             [12, 13, 14]
9           ]
10        ],
11        [15, 16, 17, 18, 19,
12          [20, 21, 22,
13            [23, 24, 25,
14              [26, 27, 29]
15            ], 30, 31
16          ], 32
17        ], 33
18      ]
```

Write a recursive function that prints all the numbers (and just numbers).

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 struct Node {
6     bool isNumber; // true if this node stores a number
7     int value;      // only used if isNumber == true
```

```

8     vector<Node> list; // only used if isNumber == false
9 };
10
11 // recursive function that prints every number from the passed in array
12 void printNums(const vector<Node>& arr) {
13     // every node in array
14     for (const Node& element : arr) {
15         // if node is a number
16         if (element.isNumber) {
17             // print it's value
18             cout << element.value << endl;
19         } else {
20             // if node is a nested list
21             printNums(element.list); // recurse into nested list
22         }
23     }
24 }
25
26 int main() {
27     vector<Node> array = {
28         { true, 1, {} },
29         { true, 2, {} },
30         { true, 3, {} },
31
32         { false, 0, { {true,4,{}}, {true,5,{}}, {true,6,{} } } },
33
34         { true, 7, {} },
35
36         { false, 0, {
37             { true, 8, {} },
38             { false, 0, {
39                 { true, 9, {} },
40                 { true, 10, {} },
41                 { true, 11, {} },
42                 { false, 0, {
43                     { true, 12, {} },
44                     { true, 13, {} },
45                     { true, 14, {} }
46                 }
47             }
48         } },
49
50         { false, 0, {
51             { true, 15, {} },
52             { true, 16, {} },
53             { true, 17, {} },
54             { true, 18, {} },
55             { true, 19, {} },
56             { false, 0, {
57                 { true, 20, {} },
58                 { true, 21, {} },

```

```

59         { true, 22, {} },
60         { false, 0, {
61             { true, 23, {} },
62             { true, 24, {} },
63             { true, 25, {} },
64             { false, 0, {
65                 { true, 26, {} },
66                 { true, 27, {} },
67                 { true, 29, {} }
68             }}
69         }},
70         { true, 30, {} },
71         { true, 31, {} }
72     }},
73     { true, 32, {} }
74 },
75
76     { true, 33, {} }
77 };
78
79 printNums(array);
80
81 }

```

Output:

```

1  anhpls@Anhs-MacBook-Pro code % g++ -std=c++17 test.cpp -o main
2  anhpls@Anhs-MacBook-Pro code % ./main
3  1
4  2
5  3
6  4
7  5
8  6
9  7
10 8
11 9
12 10
13 11
14 12
15 13
16 14
17 15
18 16
19 17
20 18
21 19
22 20
23 21
24 22
25 23

```

26	24
27	25
28	26
29	27
30	29
31	30
32	31
33	32
34	33

Video Link: <https://youtu.be/T-1W5rg-hEI>