

Name: Ryan Ellis

April 2020

Java Chapter 12 Test: Recursion

Multiple Choice Questions:

For questions 1 – 2, use the following recursive method.

```
public int question1_2(int x, int y)
{
    if (x == y) return 0;
    else return question1_2(x-1, y) + 1;
}
```

1) $1 + \text{rec}(2, 3) + 1 = 5$
 $2, 3 \rightarrow 1 + 6, 3$
 $2, 3 \rightarrow 1 + 8, 3$
 $1 + 8, 3$
 $1 + 8, 3$
 $0, 0$

1) If the method is called as question1_2(8, 3), what is returned?

- a) 11
- b) 8
- ☒ c) 5
- d) 3
- e) 24

2) Calling this method will result in infinite recursion if which condition below is initially true?

- a) $(x == y)$
- b) $(x != y)$
- c) $(x > y)$
- ☒ d) $(x < y)$
- e) $(x == 0 \ \&\& \ y != 0)$

2, 4
1, 4
0, 4
-1, 4

3) What is wrong with the following recursive sum method? The method is supposed to sum up the values between 1 and x (for instance, sum(5) should be $5 + 4 + 3 + 2 + 1 = 15$).

```
public int sum(int x)
{
    if (x == 0) return 0;
    else return sum(x - 1) + x;
}
```

- a) the exit case should return 1 instead of 0
- ☒ b) the recursive case should return $\text{sum}(x - 1) + 1$; instead of $\text{sum}(x - 1) + x$;
- c) the exit case condition should be $(x \leq 0)$ instead of $(x == 0)$
- d) the recursive case should return $\text{sum}(x) + 1$;
- e) the method should return a boolean instead of an int

$5 + \text{sum}(4)$
 $4 + \text{sum}(3)$
 $3 + \text{sum}(2)$
 $2 + \text{sum}(1)$
 $1 + \text{sum}(0)$
 0

5, 4, 3, 2, 1,

is there an error?

(cont.)

For questions 4 – 8, assume that `int[] a = {6, 2, 4, 6, 2, 1, 6, 2, 5}` and consider the two recursive methods below foo and bar.

```

public int foo(int[ ] a, int b, int j)
{
    if (j < a.length)
        if (a[j] != b) return foo(a, b, j+1);
        else return foo(a, b, j+1) + 1;
    else return 0;
}

public int bar(int[ ] a, int j)
{
    if (j < a.length)
        return a[j] + bar(a, j+1);
    else return 0;
}

```

4) What is the result of calling foo(a, 2, 9);?

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

5) What is the result of calling bar(a, 0);?

- a) 0
- b) 5
- c) 6
- d) 12
- e) 34

6) What is the result of bar(a, 8);?

- a) 0
- b) 5
- c) 6
- d) 12
- e) 34

7) Recall the Towers of Hanoi solution. If there are 6 disks to move from one Tower to another, how many disk movements would it take to solve the problem using the recursive solution?

- a) 6
- b) 13
- c) 31
- d) 63
- e) 127

For questions 8 – 9, consider the following representation of grid and the maze code from Chapter 8.

Grid:

```

1 1 1 1 1 1 0 0
0 0 1 0 0 1 0 0
0 0 1 0 0 1 1 0
0 0 1 1 0 0 1 0
0 0 0 1 1 0 0 0
0 0 0 0 1 1 1 1

```

0 1 2 3 4 5 6 7 8

Array index out of bounds?

6 + bar(1)

2 + bar(2)

9 + bar(3)

6 + bar(4)

5 + bar(5)

1 + bar(6)

6 + bar(7)

2 + bar(8)

1 1 5 31
2 3 6 63
3 7
4 15

Code:

```
public boolean traverse(int row, int column)
{
    if (valid(row, column))
    {
        boolean done = false;
        grid[row][column] = TRIED;
        if (row == grid.length - 1 && column == grid[0].length - 1)
            done = true;
        else
        {
            done = traverse(row + 1, column);
            if(!done) done = traverse(row, column + 1);
            if(!done) done = traverse(row - 1, column);
            if(!done) done = traverse(row, column - 1);
        }
        if(done) grid[row][column] = PATH;
    }
    return done;
}
```

Assume valid returns true if row and column are ≥ 0 and \leq the grid's row length or column length and the entry at this position $= 1$. And assume TRIED = 3 and PATH = 7

- 8.) If traverse is first called with traverse(0, 0); what will the first recursive call of traverse be?
- a) traverse(0, 0);
 - b) traverse(0, 1);
 - ☒ c) traverse(1, 0);
 - d) traverse(1, 1);
 - e) traverse(0, -1);.
- 9) Assume at some point in processing, grid's row 0 has become 3 3 3 1 1 1 0 0. Which direction will next be tried?
- a) up
 - b) down
 - c) left
 - ☒ d) right
 - e) none, the recursion ends at this point

10. Define the magnitude of a number as the location of the decimal point from the left of the number (that is, if a number has 4 digits followed by the decimal point, it will have a magnitude of 4). 100 would then have a magnitude of 3 and 55,555.555 would have a magnitude of 5. A partial recursive method is given below to compute a positive int parameter's magnitude. Which answer below is needed to complete the method?

```
public int magnitude(double x)
{
    if (x < 1) return 0;
    else return _____;
}
f) magnitude(x - 1) + 10;
g) magnitude(x - 10) + 1;
h) magnitude(x / 10) + 10;
☒ i) magnitude(x / 10) + 1;
j) magnitude(x / 10) * 2;
```

Short Answer:

1.) Rewrite the following iterative method as a recursive method that computes the same thing. NOTE: your recursive method will require an extra parameter. Include documentation that explains what the main program blocks do. Finally, include what the initial calling statement to the recursive method would be.

```
public int iterative1(int x)
{
    int count = 0, factor = 2;
    while(factor < x)
    {
        if (x % factor == 0) count++;
        factor++;
    }
    return count;
}
```

static

4

1

Set for bad spacing

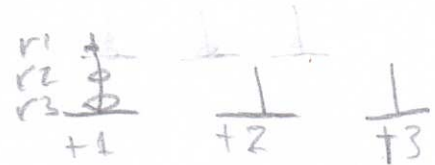
```
public static int iterative (int x, int factor) {
    int factor = 2;
    if (factor >= x) return 0;
    else if (x % factor == 0) return iterative(x, factor + 1) + 1;
    else return iterative(x, factor + 1);
}
```

- What statement that you would use to call this method?

iterative(x, 2);

2. Describe how to solve the Towers of Hanoi problem using 3 disks.

- write down, move for move, how to solve the problem
- In what way is this algorithm recursive?
- Give the general recursive solution (involving n rings).
- Give the exit case



1) r1 → +3 2) r2 → +2 3) r1 → +2 4) r3 → +3
5) r1 → +1 6) r2 → +3 7) r1 → +3

it is recursive since the previous n's tower must be built in order for the next n tower to be made

Solution: $n_{next} = 2n_{prev} + 1$

exit case: +1 and +2 have no rings

3. Assume a function $g(x)$ is defined as follows where x is an int parameter:

$g(x) = g(x-1) * g(x-3)$ if x is even and $x > 3$
 $= g(x-2)$ if x is odd and $x > 3$
 $= x$ otherwise

Write a recursive method to compute g .

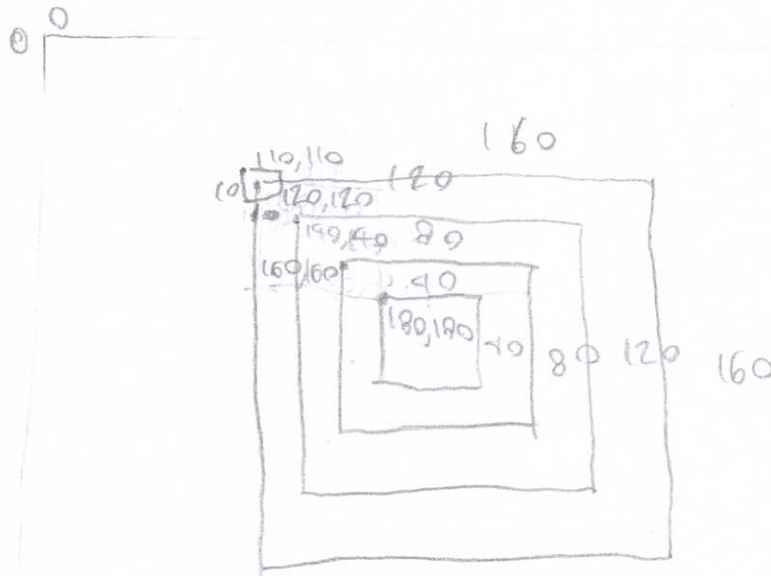
```
public static int g(int x) {
    if (x > 3) {
        if (x % 2 == 0)
            return g(x-1) * g(x-3);
        else return g(x-2);
    }
}
```

else return x;

3

4. If the following method is part of a JavaFX program where all the other expected graphic objects are taken care of, and it is first called with `addRect (40)`, show what is displayed, being clear to label the rectangles' top left vertices (e.g. "(12, 12)") and the width and height of each rectangle.

```
public void addRect (int x)
{
    if (x >= 200)
        getChildren().add(new Rectangle (110, 110, x/20, x/20));
    else
    {
        Rect newRect = new Rectangle (200 - x/2, 200 - x/2, x, x)
        newRect.setStroke (Color.black);
        newRect.setStrokeW (1);
        newRect.setFill (null); // outline only
        getChildren().add(newRect);
        addRect(x + 40);
    }
}
```



5. The following drawing is a line using the Koch snowflake design where order = 2. Show how it would appear with order = 3.



6. Either:

- A. Make up a joke (or modify one that you already know) having to do with, or showing, recursion, and explain specifically why it is recursive (it is not enough to show that there is repetition)

. . . OR . . .

- B. Design from scratch a recursive fractal that has not yet been discussed or created in our class. Show the first three levels of the fractal, tell what parameters would need to be sent to the drawing method, and explain how the fractal is recursive.

A. Do you understand recursion yet?
if you don't, read this again.

Ex: You are trapped in the recursion until you reach the exit case; understanding recursion every time you read it, you bring your knowledge of recursion from the previous iteration.