

Name Tanmai Karisipudi Period 1

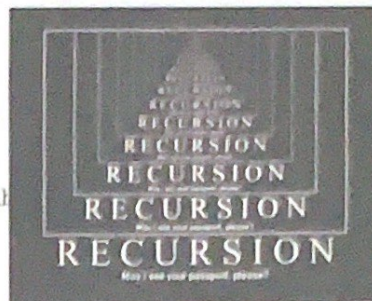
Recursion Exercises

Practice #1

Use the following method puzzle to answer problems 1 - 3.

/* precondition: base and limit are nonnegative numbers */

```
public int puzzle(int base, int limit)
{
    if ( base > limit )
        return -1;
    else if ( base == limit )
        return 1;
    else
        return base * puzzle(base + 1, limit);
}
```



1. Identify the base case(s) of method puzzle if base > limit, if base = limit
2. Identify the recursive case(s) of method puzzle if base < limit it becomes recursive
3. Show what would be displayed by the following calls.
 - a) System.out.println(puzzle(14,10)); -1
 - b) System.out.println(puzzle(4,7)); 120
 - c) System.out.println(puzzle(0,0)); 1

In problems 4 - 6, show the output that will be displayed by the call show(123);

4.

```
public void show ( int n )
{
    if ( n > 0 )
        show (n/10);
    System.out.print (n%10);
}
```

0123

5.

```
public void show ( int n )
{
    System.out.print (n%10);
    if (n>0)
        show (n/10);
}
```

3210

6.

```
public void show ( int n )
{
    System.out.print (n%10);
    if (n>0)
        show (n/10);
    System.out.print (n%10);
}
```

0123 3210

7. Complete the code to recursively evaluate the sum: $sum = 1 + 1/2 + 1/3 + \dots + 1/n$, $n \geq 1$.

```
public double sum(int n) // n>=1
{
    if ( n==0 )
        return 1;
    return 1.0/n + sum(n-1);
}
```

8. What does this code count?

```
public int count (int n)
{
    if (n==1)
        return 1;
    return 1 + count(n-1);
}
```

It just counts the number so it will just return n.

Yannai Gal-rivlin

Recursion Exercises

Practice #2

Evaluate each recursive function `foo` for the given parameter values.

- ```
//x is nonnegative
public int foo (int x){
 if (x == 0)
 return (x + 1);
 return 1 + foo (x-1);
}
```

`foo(3)` 4
- ```
public int foo (int x, int y)
{
    if (x <= 0)
        return (0);
    if (y >= x)
        return 1 + foo(y,x);
    return 2 + foo(x-3,y-1);
}
```

`foo(6,5)` 8
- ```
public int foo(int x)
{
 if (x < 5)
 return x*x + 1;
 if (x == 5)
 return x*x - 3;
 return foo(x-2);
}
```

`foo(foo(foo(foo(foo(foo(3))))))` 17

Show the output of each recursive method.

- ```
public void p(int n)
{
    p(n - 1);
    System.out.print (n);
    p(n - 1);
}
...
p(3);
System.out.println();
```

?
- ```
int[] list = new int[10];
// list contains { 4, -1, 5, 1, 8, 3, -2, 1, 6, 7 };

public int calc(int[] list, int first, int last)
{
 if (first > last)
 return 0;
 return list[first] + calc(list,first+1,last);
}
...
System.out.println("Result:" + calc (list, 2, 7));
```

Result: 16

6. What happens when you call `doSomething(3)`?

```
public static void doSomething(int value){
 if (value > 0){
 doSomething(value-1);
 doSomething(value-1);
 System.out.print(" " + value);
 }
}
```

1 1 2 1 1 2 3