AP Computer Science A
Ticket Out the Door
Set 27: Recursion

Name _____ Period _____

Skill 27.3 Exercise 1

Complete the stack diagram for the code block shown, then indicate the output.

```
class Main {

public static void main(String[] args) {
          Recursion.reduceByOne(4);
     }
}

class Recursion{

public static void reduceByOne(int n){
          if(n > 0)
               System.out.println(n);
               reduceByOne(n-1);
               System.out.println(n);
          }
     }
}
```

Stack	Output

Skill 27.4 Exercise 1

Complete the stack diagram for the code block below.

Consider the following method.

```
public static int calcMethod(int num)
{
  if (num == 0)
  {
    return 10;
  }
  return num + calcMethod(num / 2);
}
```

Complete the stack diagram for the following call, then indicate what is returned.

```
calcMethod(16)
```

Stack	Output

AP Computer Science A
Ticket Out the Door
Set 27: Recursion

Name _____ Period _____

Skill 27.4 Exercise 2

Complete the stack diagram for the code block below.

Consider the following method.

```
public String goAgain(String str, int index)
{
  if (index >= str.length())
    return str;

  return str + goAgain(str.substring(index), index + 1);
}
```

Complete the stack diagram for the following call, then indicate what is printed.

```
System.out.println(goAgain("today", 1));
```

Stack	**Output**

Skill 27.4 Exercise 3

Complete the stack diagram for the code block shown, then indicate the output.

```
class Main {
public static void main(String[] args) {
          System.out.println(Recursion.pls(4));
     }
}
class Recursion{

     public static int pls(int n)
     {
          if (n == 0)
               return 5;
          else if (n == 1)
               return 11;
          else
               return pls(n - 1) + 2 * pls(n - 2);
     }
}
```

AP Computer Science A
Ticket Out the Door
Set 27: Recursion

Name _____ Period _____

Stack
Output

Skill 27.4 Exercise 4

Complete the stack diagram for the code block shown, then indicate the output.

```
class Main {
public static void main(String[] args) {

          Recursion.homer(9));
     }
}

class Recursion{
public static void homer(int n)
{
     if (n <= 1)
          System.out.print(n);
     else
          homer(n / 2);

System.out.print(",” + n);

}
```

Stack	**Output**