# **Recursion Exercises**

Write the following methods. It will enrich your learning experience if you can test your methods and make sure that they work.

1. Write a method called letters that takes a lower case letter as its argument and displays the sequence of letters from ‘a’ to the given letter.

|  |  |  |  |
| --- | --- | --- | --- |
| *For example* | letters(‘a’); | *displays* | a |
|  | letters(‘d’); | *displays* | abcd |

public static void displayLetters (String targetLetter, String currentLetter)

{

if (currentLetter == targetLetter)

{

System.out.print(current);

}

else

{

System.out.print(current);

displayLetters(targetLetter, ++currentLetter);

}

}

2. Write a method called twos that takes a single integer as its argument and returns the number of factors of 2 in the number. (Hint: odd numbers have no factors of 2, numbers that are twice an odd number have one, numbers that are four times an odd have two, and so on.)

|  |  |  |  |
| --- | --- | --- | --- |
| *For example* | System.out.println(twos(-12)); | *returns* | 2 |

public static int twos(int num)  
{  
 if ((double)num % 2 == 1)  
 {  
 return 0;  
 } else if ((double)num % 2 == -1)  
 {  
 return 0;  
 } else  
 {  
 return 1 + *twos*(num / 2);  
 }  
}

3. Write a boolean method called powerof3 that takes a single positive integer argument and returns true iff the integer is a perfect power of 3 such as 1, 3, 9, 27, 81, …

|  |  |
| --- | --- |
| *For example* | if (powerof3(81))  System.out.println(81+“ is a power of 3.\n”);  else  System.out.println(81+“ is not a power of 3.\n”); |
|  |  |
| *Displays* | 81 is a power of 3. |

public static boolean powerOf3(int num)  
{  
 if (num % 3 == 0 && num < 9)  
 {  
 return true;  
 } else if (num % 3 != 0 && num < 9)  
 {  
 return false;  
 } else  
 {  
 return *powerOf3*((num / 3) + num % 3);  
 }  
}

4. Write a function called reverse that takes a single long integer argument and returns the result of reversing its digits. For partial credit, write a void function that just displays the number in reverse.

|  |  |  |  |
| --- | --- | --- | --- |
| *For example* | System.out.println(reverse(-12)); | *returns* | -21 |
|  | System.out.println(reverse(123456)) | *returns* | 654321 |

public static void reverse(int num)  
{  
 if (num % 10 == 0)  
 {  
 return;  
 } else  
 {  
 System.*out*.print(num % 10);  
 return *reverse*( num / 10);  
 }  
}

5. Write a function called base5 that takes a single nonnegative integer argument and displays its base five equivalent.

|  |  |  |  |
| --- | --- | --- | --- |
| *For example* | base5(136); | *displays* | 1021 |
|  | base5(5); | *displays* | 10 |

public static void base5(int num)  
{  
 if (num == 0)  
 {  
 System.*out*.println();  
 return;  
 } else  
 {  
 *base5*(num / 5);  
   
 System.*out*.print(num % 5);  
 }  
}

**6.**  Write a function called printWithCommas that takes a single nonnegative long integer argument and displays it with commas inserted properly.

|  |  |  |  |
| --- | --- | --- | --- |
| *For example* | printWithCommas(12045670); | *displays* | 12,045,670 |
|  | printWithCommas(1); | *displays* | 1 |

public static void printWithCommas(int num)  
{  
 if (num > 999)  
 {  
 *printWithCommas*(num/1000);  
 System.*out*.print(',');  
 System.*out*.print(num%1000);  
 }  
 else  
 {  
 System.*out*.print(num);  
 }  
}