2. This program would not work because the recursion would never end. Since the base case is y==0, it would be necessary to include some type of decrease in y to get to that point so the program doesn’t run forever. Instead, on each run through the program adds 1 to y, so with the invocation of pow(2,3), y would forever increase and there would be too many recursive sections that would never end. The result of pow(2,3) is actually an error, a StackOverflowError. According to the javadocs this error is caused when “an application recurses too deeply.” In this situation the program would check if y is 0, and since it isn’t it would return x times the method with x and y+1. Using this y would never become 0 so the base case would never be met.

3.

String reversei(String str) {  
 **char**[] temp = str.toCharArray();  
 String str2 = **""**;  
 **for** (**int** i=temp.**length**-1; i>=0; i--) {  
 str2 += temp[i];  
 }  
 **return** str2;  
}

4.

**int** min(**int**[] a, **int** index) {  
 **if** (index == a.**length** - 1)  
 **return** a[index];  
 **int** min = *min*(a, index + 1);  
 **if** (a[index] < min)  
 **return** a[index];  
 **else  
 return** min;  
}

5.