**Ch.19 Recursion Review Questions and Exercises**

**Short Answer**

1. What is the base case of each of the recursive functions listed in questions 12, 13, and 14?

2. What type of recursive function do you think would be more difficult to debug, one

that uses direct recursion, or one that uses indirect recursion? Why?

3. Which repetition approach is less efficient, a loop or a recursive function? Why?

4. When should you choose a recursive algorithm over an iterative algorithm?

5. Explain what is likely to happen when a recursive function that has no way of stopping

executes.

**Fill-in-the-Blank**

6. The \_\_\_\_\_\_\_\_\_\_ of recursion is the number of times a function calls itself.

7. A recursive function’s solvable problem is known as its \_\_\_\_\_\_\_\_\_\_. This causes the

recursion to stop.

8. \_\_\_\_\_\_\_\_\_ recursion is when a function explicitly calls itself.

9. \_\_\_\_\_\_\_\_\_ recursion is when function A calls function B, which in turns calls function A.

**Algorithm Workbench**

10. Write a recursive function to return the number of times a specified number occurs in

an array.

11. Write a recursive function to return the largest value in an array.

**Predict the Output**

What is the output of the following programs?

12. #include <iostream>

using namespace std;

int function(int);

int main()

{

int x = 10;

cout << function(x) << endl;

return 0;

}

int function(int num)

{

if (num <= 0)

return 0;

else

return function(num − 1) + num;

}

13. #include <iostream>

using namespace std;

void function(int);

int main()

{

int x = 10;

function(x);

return 0;

}

void function(int num)

{

if (num > 0)

{

for (int x = 0; x < num; x++)

cout << '\*';

cout << endl;

function(num − 1);

}

}

14. #include <iostream>

#include <string>

using namespace std;

void function(string, int, int);

int main()

{

string mystr = "Hello";

cout << mystr << endl;

function(mystr, 0, mystr.size());

return 0;

}

void function(string str, int pos, int size)

{

if (pos < size)

{

function(str, pos + 1, size);

cout << str[pos];

}

}