Data Structures
Fall 2020
Prof. George Wolberg
Midterm (Part 1: Chap. 1-3)

CHAPTER 1 PROBLEMS

Problem 1.1. (1 point)

Convert each time formula to big-O notation.
a) 10n
b) $2n^2$
c) 3log(n)
d) $2n^2$ + 10n

Problem 1.2. (1 point)

List the following formulas in order of running time analysis, from greatest to least time requirements, assuming that n is very large: n^2; 50 log n; 1,000,000; 10n + 10,000.

Problem 1.3. (1 point)

Write the simplest big-O expression to describe the number of operations required for the following algorithm:

```
for (i = 1; i < N; ++i) {
   ...statements that require exactly i operations...
}
```

Problem 1.4. (1 point)

What will happen if a function is executed and the precondition for the function is not met?
A. An error message will be printed.
B. The program will loop indefinitely.
C. The system will crash.
D. Any of the above results could happen.

Problem 1.5. (1 point)

If the precondition fails, it is a good idea to write a useful error message and then halt the program. Why is the program halted?
A. Most operating systems forbid continuation.
B. The function is no longer guaranteed to make the postcondition true.
C. The function's memory requires have become exponential (or worse).
D. The function's running time has become exponential (or worse).

Problem 1.6. (1 point)

Which of these is used to stop the program execution when a precondition is not met.
A. assert();
B. exit();
C. return();
D. void();

Problem 1.7. (1 point)

What is the easiest way to turn off all assertion checking in a program?
A. #define NASSERT should appear before any include directives.

B. #define NDEBUG should appear before any include directives.
C. Comment out each occurrence of the assert statement.
D. All of the above.

Problem 1.8. (1 point)

What does a run-time analysis usually count?
A. The number of arithmetic and other operations required for the program to run
B. The number of megabytes required for the program to run
C. The number of seconds required for the program to run
D. The number of seconds plus the number of megabytes
E. The number of seconds times the number of megabytes

Problem 1.9. (1 point)

Which of these is the correct big-O expression for 1+2+3+...+n?
A. O(log n)
B. O(n)
C. O(n log n)
D. O(n^2)

Problem 1.10. (1 point)

Prove your answer for the previous problem.

Problem 1.11. (1 point)

Which of the following formulas in big-O notation best represent the expression n^2+35n+6?
A. O(n^3)
B. O(n^2)
C. O(n)
D. O(42)

Problem 1.12. (1 point)

Here is some code for an integer variable n:

```
while (n > 0) n = n/10; // Use integer division
```

What is the worst-case time analysis for the above loop?
A. O(1)
B. O(log n)
C. O(n)
D. O(n^2)

CHAPTER 2 PROBLEMS

Problem 2.1. (1 point)

What is the difference between a class and an object?

Problem 2.2. (1 point)

Here is part of the throttle declaration:

```
class throttle
{
public:
   void shut_off();
   double flow();
   ...
private:
   int position;
};
```

Write several lines of C++ code to declare a throttle called quiz, activate the member function that shuts off quiz's flow, and then print the current flow from quiz.

Problem 2.3. (1 point)

What is a macro guard in a header file, and what is its purpose? Give a small example as part of your answer.

Problem 2.4. (1 point)

Write one clear sentence telling me when it would be appropriate to use a reference parameter.

Problem 2.5. (1 point)

Is it possible for a member function of a class to activate another member function of the same class?
A. No.
B. Yes, but only public member functions.
C. Yes, but only private member functions.
D. Yes, both public and private member functions can be activated within another member function.

Problem 2.6. (1 point)

Can two classes contain member functions with the same name?
A. No.
B. Yes, but only if the two classes have the same name.
C. Yes, but only if the main program does not declare both kinds
D. Yes, this is always allowed.

Problem 2.7. (1 point)

What is the primary purpose of a default constructor?
A. To allow multiple classes to be used in a single program.
B. To copy an actual argument to a function's parameter.
C. To initialize each object as it is declared.
D. To maintain a count of how many objects of a class have been created.

Problem 2.8. (1 point)

Suppose that the foo class does not have an overloaded assignment operator. What happens when an assignment a=b; is given for two foo objects?
A. The automatic assignment operator is used
B. The copy constructor is used
C. Compiler error
D. Run-time error

Problem 2.9. (1 point)

When should you use a const reference parameter?
A. Whenever the data type might be many bytes.
B. Whenever the data type might be many bytes, the function changes the parameter within its body, and you do NOT want these changes to alter the actual argument.
C. Whenever the data type might be many bytes, the function changes the parameter within its body, and you DO want these changes to alter the actual argument.
D. Whenever the data type might be many bytes, and the function does not change the parameter within its body.

Problem 2.10. (1 point)

Here is a small function definition:

```
void f(int i, int &k)
{
   i = 1;
   k = 2;
}
```

Suppose that a main program has two integer variables x and y, which are given the value 0. Then the main program calls f(x,y); What are the values of x and y after the function f finishes?
A. Both x and y are still 0.
B. x is now 1, but y is still 0.
C. x is still 0, but y is now 2.
D. x is now 1, and y is now 2.

CHAPTER 3 PROBLEMS

Problem 3.1. (1 point)

For the bag class in Chapter 3 (using a fixed array and a typedef statement) what steps were necessary for changing from a bag of integers to a bag of double values?
A. Change the array declaration from
int data[CAPACITY] to double data[CAPACITY] and recompile.
B. Change the int to double in the typedef statement and recompile.
C. Round each double value to an integer before putting it in the bag.
D. Round each double value to an integer after taking it out of the bag.

Problem 3.2. (1 point)

I have an array named data, which contains n integers. I want to print all of the numbers, starting at data[0]. BUT if the number 42 occurs, then I want to stop my printing just before the 42 (without printing the 42!). Here is most of my for-loop to accomplish my goal:

```
for (i = 0; _____; i++)
    cout << data[i] << endl;
```

What is the correct way to fill in the blank? (If there is more than one correct answer, please select E.)
A. (data[i] != 42) && (i < n)
B. (data[i] != 42) || (i < n)
C. (i < n) && (data[i] != 42)
D. (i < n) || (data[i] != 42)
E. More than one of the above answers is correct.

Problem 3.3. (1 point)

What is the best C++ statement to use when a program must choose between several alternatives that are controlled by the value of a single variable? A. do-while statement B. for-statement C. if-else statement D. switch statement E. while statement