Quiz/Homework Assignment #1
CSE326
Homework due date:
Wednesday, January 12th, beginning of class

January 10, 2000

This quiz is your first homework assignment. Any question you answer correctly in class will be considered correct for the purposes of the assignment. Any question you do not answer or which you answer incorrectly becomes a short answer question for the homework assignment. We will hand back the graded quiz on Monday.

Good luck! Make sure that you bubble your answers in the corresponding space on the answer sheet, and keep this question sheet so that you can work on any questions which you know you missed over the weekend.

1. $\Sigma_{i=0}^{\infty} x^i = 1/(1-x)$ is true if and only if:
   (a) $-1 < x < 1$
   (b) $x \neq 1$
   (c) $0 < x < 1$
   (d) $x > 1$
   (e) $x = 1/2$

2. *Note: this question was removed from the quiz and homework.*

   Which of the following formulas has the **smallest** value?
   (a) $(\Sigma_{i=1}^{5} i)(\Sigma_{i=1}^{5} i)$
   (b) $\Sigma_{i=0}^{100} (-5)^i$
   (c) $\Sigma_{n=0}^{5} (\Sigma_{i=0}^{n} 2^i)$
   (d) $\Sigma_{i=0}^{10} 2^3$

3. Which of the following formulas has the **greatest** value?
   (a) $log_{10} 1024$
   (b) $(log_{13} 343)/(log_{13} 7)$
   (c) $log_2 (1/128)$
   (d) $log_8 256$
   (e) $(log_a a^2) + (2 log_a \sqrt{a})$

4. Which of the following is not equal to $a^3 + 4ab^2$?
   (a) $a(a^2 + \frac{6ba}{3(ab)^0}^2)$
   (b) $\frac{a^4 b}{ab} + a(2b)^2$
   (c) $((2^{1+a+b})(2^{1+a-b})a^2b^2 + a^4)/(a4^a)$
   (d) $4 \frac{((2ab)^2 + a^4)}{4a}$
   (e) none of these

5. *Note, this question has two correct answers. If you got either right on the quiz, you were correct. Otherwise, find and justify both for the homework.*

   Which of the following does **not** equal 1?
   (a) $5 mod 2$
   (b) $98179387171 - 19873956961 mod 10$
   (c) $4879165163 * 91991931877 mod 10$
   (d) $1987917135 + 19871938574 mod 2$
   (e) $1 mod 1999 + 1$

6. What is the best way to implement a queue?
   (a) A linked list, because the memory usage is proportional to the number of elements in the queue.

(b) A circular array, because no memory is wasted with next (or previous) pointers.
(c) A linked list, because it doesn't limit how many elements you can have in the queue.
(d) A circular array; it's faster because you don't have to deference next or previous pointers.
(e) It depends.

7. *Note, this question has* no *correct answer on the quiz. You should find and justify the correct answer for homework.*

   The following pseudocode uses the functions C, F, and G.

```
if (C()) {
  for i = 1 to n
    for j = 1 to n
      F()
} else {
  F()
  G()
}
```

   In terms of the runtime of C, F, and G ($T(C)$, $T(F)$, and $T(G)$, respectively) the asymptotic bound on the runtime of this code must be equivalent to which of the following?

   (a) $O(n^2)$
   (b) $O(n^3T(F) + T(C)T(F)T(G))$
   (c) $O(2nT(C)T(F) + T(C)T(F)T(G))$
   (d) $O(n^2T(F)^2T(G))$
   (e) $O(n^2T(C)T(F) + T(C)T(F)T(G))$

8. Given that $T_1(N) \in \theta(f(N))$ and $T_2(N) \in \theta(f(N))$ which of the following is **not** true?

   (a) $T_1(N) + T_2(N) \in O(f(N))$
   (b) $T_1(N) - T_2(N) \in o(f(N))$
   (c) $T_1(N)/T_2(N) \in O(1)$
   (d) $T_1(N) \in O(T_2(N))$
   (e) $T_1(N) \in \Omega(T_2(N))$

9. Which of the choices below is a correct upper bound on the running time of this code?

```
sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < i; j++)
    sum++;
```

   (a) $O(1)$
   (b) $\omega(n^2)$
   (c) $o(n^2)$
   (d) $O(n^2logn)$
   (e) none of these

10. Under which conditions would you want to store a reference to each row and column header in each element of a cross list?

   (a) Time is cheap, memory is precious.
   (b) Memory is precious, but the actual information stored at each element is small.
   (c) Time is precious, memory is cheap, but the actual information stored at each element is large.
   (d) none of the above

11. What is a tight upper bound on the running time of this code?

```
for (i = 1; i * i <= n; i++)
  if (n % i = 0)
    cout << i << ' ';
cout << endl;
```

   (a) $O(\sqrt{n})$
   (b) $O(1)$
   (c) $\Omega(n)$
   (d) $O(n^2)$
   (e) none of the above

12. Which choice below is a tight asymptotic bound for a function with the following runtime characteristics?

   $T(n) = T(n-1) + n$
   $T(1) = 1$

   (a) $\Omega(n)$
   (b) $\theta(n)$

(c) $O(n^3)$
(d) $\theta(n^2)$
(e) none of the above

13. Given the following templated class:

```
template <class T>
class Templ
public:
  T min (T x, T y) {
    if (x < y) return x;
    else return y;
  }
};
```

What can we say about class T?

(a) It has a less than test function, 'operator <'.
(b) It has an equality testing function, 'operator =='.
(c) It has a less than test function, 'less'.
(d) All of the above.
(e) None of the above.

14. What is the flaw in the following inductive proof?

Theorem All dogs are the same color.

Proof Proof by induction. Formally, we wish to prove that, for any set of n dogs, all dogs are the same color.

Base case: $n = 1$ Clearly, if we have a set of only one dog, the dog is the same color as itself.

Induction step We assume that the property is true for some n, and show that it is true for $n+1$. We select two sets of n dogs A and B such that all of the $n+1$ dogs occurs in at least one of the sets A or B, And there is one dog that appears in both sets (call this dog "Rover"). Since A and B are of size n, we know from the Induction Hypothesis that the dogs in each set are the same color. Moreover, since Rover is in both sets, both sets must be the same color. So, the property holds for $n + 1$ dogs. QED.

(a) The base case does not cover sets with no dogs ($n = 0$), so we can't conclude that all dogs are the same color for any number of dogs.
(b) We did not mention what happens for $n > 1$, so we have not proven the property for any number of dogs.
(c) The given Induction Step does not work when $n = 1$, so the property does not hold for larger n.
(d) None of the above.

15. In C, strings are implemented as arrays. They could be implemented using linked lists, specifically a linked list of characters. In the string "Data Structures are supercool", the head of the list would be the letter 'D', the next node would be 'a', the next node would be 't', etc.

We could use a singly linked list or a doubly linked list. For which of the following operations, would you want to use a doubly-linked list (assuming the program performed this operation frequently)? Assume that only big-O time is significant.

(a) Find character (e.g., does some string contain the letter 'A'?)
(b) Count length of string (e.g., given "Data", the length is 4)
(c) Reverse string (e.g., given "Data", output "ataD")
(d) Find previous (i.e., given a pointer to a character, find the previous character)
(e) Concatenate strings (e.g., given "Data" and "Structures", output "DataStructures")