CS 216 Exam 1 – Fall 2002

Name _______________________  Section _______________________

Email Address _________________  Student ID # _________________

This exam is closed note, closed book. You will have an hour and fifty minutes total to complete the exam. You may NOT use calculators.

Good Luck!

<table>
<thead>
<tr>
<th></th>
<th>MAX</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Define and give an example of each:

1. a) (2 points) length of a path in a tree
   
   b) (2 points) AVL tree structure property
   
   c) (2 points) normalized (in the context of floating point values)
   
   d) (2 points) High level language
   
2. (2 points) Give an example of a FIFO data structure and give the names of the operations used to add and remove elements from the structure:

3. (2 points) Give two examples of things that may have an effect on running time, but Big-O time comparison ignores.
4. [2 points] A pixel in the cs216 graphics library can be any one of 700 possible colors. You are charged with creating a data structure to represent a pixel. What is the \textit{minimum} number of bits needed to represent the color of a pixel?

5. [6 points total] In the highly impractical cs216 architecture, words are 9 bits long.
   a) Show how the number 13 would be represented in twos complement in a 9-bit word.

   b) Show how -5 would be represented in twos complement in a 9-bit word.

   c) Show how the result of adding 13 and -5 would be represented in a 9-bit word.

6. [4 points] Write the following infix expression as a postfix expression:
   \[ a + ((b \div c) * \left( d + e \right)) - f \]
7. (8 points total) Describe the running time of the following pseudocode in Big-Oh notation in terms of the variable \( n \). *Show your work for partial credit.*

a) ```
int test(int k) {
    for (int i = 0; i < k; ++i) {
        if (i < 10) {
            cout << "too small";
        }
    }
    return k;
}

a = test(n);
``` 

b) ```
for (int i = 0; i < 99; ++i) {
    if (i > 50) {
        for (int j = 0; j < i; ++j) {
            cin >> b;
            a = a + b;
        }
    }
}
``` 

c) ```
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < i * i; ++j) {
        sum = sum + i;
        for (int k = 0; k < j * j; ++k)
            a[k] = a[k] + sum;
    }
}
``` 

d) ```
for (int j = 0; j < n; ++j) {
    b = a * c;
    for (int i = 0; i < j * n; ++i)
        a = a + b;
    for (int k = 0; k < n; ++k)
        c = b + c;
}
8. (6 points total) What is the representation of each of the following in the indicated radix? Be sure to show your work.

a) $241_5$ in decimal

b) $2122_3$ in hex

c) $4A7_{11}$ in radix 10

9. (6 points) Consider the positive binary integer represented in two’s complement: $0101101101010000_2$.

a) Express this binary number in octal

b. Express this binary number in hexadecimal

c. Negate the number (i.e. give the two’s complement representation of a negative version of the same number) Use the same number of bits.
10. a) (3 points) Draw the binary search tree created by inserting these values in this order:

```
4 3 0 9 8 6 2 5 1 7
```

Draw T.

b) (2 points) Give a pre-order traversal of your tree shown above:
11. (3 points each) The cs216list class is implemented as a doubly linked list (similar to the one you used in lab #1). For each operation give 1) How you would most efficiently implement the operation in the cs216list, 2) WHAT exactly is the worst case scenario and 3) what is the worst case Big-Oh running time of this scenario. State any assumptions you make about the structure of a cs216list or the implementation of its operations.

a) Reading an element of a cs216 with the [ ] operator.

b) `push_back(T val)` – inserts a copy of val after the last element in the cs216list.

c) `size()` – returns the size of the cs216list.

d) `insert(iterator pos, T val)` – inserts a copy of val prior the element in the cs216list referred to by pos.
12. (3 points) Given the following tree:

```
     20
    /   \
  13     77
 /     /  \
2  15  78
  /          \
14  19
```

Is it an AVL tree? If not, circle the node(s) where the AVL property is violated. Why or why not (must answer for any credit)?
13. (5 points) In C++ it is possible to declare a function as follows:

```
int test (int a[][100]) {
    int c;
    c = a[3][6];
    return c;
}
```

a) (3 points) What does the code the compiler generates for line 3 look like? (pseudo-code is o.k.)

b) (2 points) If we changed the parameter \texttt{a} to a 3-dimensional array, which dimensions of the array must be specified? (give an example of the function declaration (line 1) for the new function) Why will this new declaration work?
14. (7 points) Assume we are using the 32-bit IEEE single precision floating point format as
described in class and used in lab. The mantissa has 24 bits including the hidden bit, there is one
sign bit, and there are eight exponent bits. The exponent is stored in excess 127.

What decimal floating point number is represented by the following 32 bits? SHOW
YOUR WORK!

0011 0000 1001 1000 0000 0000 0000 0000

a) Is this a positive or negative number?

b) What is the exponent (in base 10)?

c) What is the value of the mantissa (in base 10)

d) What is the total value?

Note: you may leave your answer in the form: $value_{10} \times base^{exponent}$

Where you specify value, base and exponent.
15. (24 points total) a) (10 points) Implement a stack ADT in C++. The underlying representation of the stack should be an array. Your stack should store integers and should handle errors. You will be graded mostly on the correctness of the ideas of your solution rather than exact C++ syntax, but your solution should be clear. Correct C++ code is the best way to ensure we understand your solution.

You should implement the following:

Stack class declaration
Stack class constructor

You should implement the following member functions:

push(int val) - pushes val onto the stack
top() - returns the value on the top of the stack (without removing it from the stack)
pop() - returns the value on the top of the stack and removes it from the stack
isEmpty() - returns true if the stack contains no elements, false otherwise
isFull() - returns true if the stack is full, false otherwise
size() - returns the number of elements currently in the stack
[extra space for use in part a) or b)]
b) (9 points) [NOTE: do it as pass by reference? Otherwise S is pass by value and in most cases won’t be changed. Also, make it clear that they cannot add more member functions to the Stack class to access private data members directly!]
Implement a non-member function `print_stack(stack S)` in C++.
`print_stack` should print out all elements currently in the stack in the order they were originally inserted into the stack. For example:

```
S.push(1);
S.push(2);
S.push(3);
print_stack(S);
```

Should print: 1 2 3

`print_stack` cannot access private data members and functions of the stack class. When you return from `print_stack`, the original stack should be unchanged.
c) (3 points) Give the worst case big-O running time of your implementation for each of the following operations (in terms of n where n is the number of elements currently in the stack). For partial credit, explain briefly how you got your answer.

push(int val)

pop()

size()

d) (2 points) Give the worst case big-O running time of your implementation of print_stack (in terms of n where n is the number of elements currently in the stack). For full credit you must explain briefly how you got your answer.

On my honor, I have neither given nor received unauthorized aid on this exam.

_________________________
Sign your name