

Accepted: on paper, in lecture (1-1:50pm) on Wednesday, **June 3rd** only.
No email submission allowed.

Name One: (4 pts) _____ UCSBNetID (4 pts) _____

Section (2 pts) Circle one: Thu 4pm Fri 10am Fri 11am Fri noon

1. Read (or re-read) the section from the book on binary numbers—p. 137-139.

You may also want to review your notes on binary numbers from last Friday's lecture (05/29)

a. (4 pts) Convert 17 from decimal to binary

b. (4 pts) Convert 00111001 from binary to decimal

- As a reminder, leading zeros can be ignored—but it is traditional to write binary in blocks of 8 bits.
- So, in this case 00111001 is the same as 111001

c. (4 pts) Convert 1010 0111 from binary to decimal

- **Note:** Another tradition when writing groups of bits is to leave a space every three bits or every four bits, counting from the right.
- So, the number 1010 0111 is the same as the number 10100111
- This is similar to the way we put commas into large numbers every three spaces, starting from the right.
- For example, the commas aren't part of the number when we write:
 - 2008 estimated population of LA County: 10,363,850
- In the same way, 01 1000 0110 or 0 110 000 110 are equivalent ways of writing 0110000110

d. (4 pts) Convert 1010 0111 from binary to hexadecimal

- **Reminder:** to convert from binary to hex, you convert each group of 4 bits (starting from the right) to a decimal number by adding up $8+4+2+1$ (using only the numbers where the bit is 1, and leaving out the ones where the bit is a zero.) and then write down the corresponding digit (unless it is 10-15, in which case you write down A-F, respectively.)
- **Example:** 1110 0101 corresponds to E5, because 1110 gives us $8+4+2 = 14$ (E), and 0101 gives us $4+1 = 5$.

e. (4 pts) Convert 001 011 101 from binary to octal

- **Reminder:** converting from binary to octal works just like binary to hex, except the groups are groups of three, so the digits end up being between 0 and 7. You are adding up $4+2+1$ for each group of three bits, choosing only the numbers where the bit is a 1.
- **Example:** 110 111 100 corresponds to 674 in octal, because 110 gives us $4+2 = 6$, and 111 gives us $4+2+1 = 7$, and 100 is just 4.

Now please turn over for more problems

- (10 pts) Remember what we said about the "stack" during Friday's lecture—that each time a method is called, a "call frame" goes on the stack, and each time a method returns, that call frame is removed from the stack.

Look at the code on p. 591, and consider what would happen if we removed lines 25 and 26 (these are called "base cases").

What *would* happen if we removed these lines, and then called the method on a triangle of size 3?

(Hint: you can think this through—but if you read carefully, the answer is somewhere in Section 13.1. Rather than just hunting for the answer and writing it down, I'd encourage you to try to think it through on your own first—you'll learn more than way, and be better prepared for the final exam. *Then* hunt for the answer to make sure you thought it through correctly.)

- (10 pts) Consider a method that returns a string of X's of length n, e.g.

```
StringUtils.nTimesX(3) returns "XXX"  
StringUtils.nTimesX(0) returns "" (empty string)  
StringUtils.nTimesX(5) returns "XXXXX"
```

Suppose you were going to write this function using recursion. Read through the section on pages 597-600 called "Thinking Recursively". Then, see if you can fill in the blanks below.

If you want to solve this by actually programming it, you can download the code with this link:

<http://www.cs.ucsb.edu/~pconrad/cs10/09S/homework/H16/H16Project.zip>

```
public class StringUtils  
{  
    public static String nTimesX(int n)  
    {  
        if (n <= 0)  
            return _____ ;  
        else  
            return "X" + _____ ;  
    }  
}
```

End of H16