

Problem 1: Textbook Questions [4 marks]

Answer the following questions from Fluency with Information Technology.

Chapter 3, Short Answer #8: A company that supplies connections to the Internet is called a(n):

Chapter 3, Short Answer #11: Special computers that send files to Web browsers elsewhere on the Internet are known as:

Chapter 3, Multiple Choice #7: The part of an email address to the right of the @ is most like a

- A. mailbox
- B. post office
- C. letter
- D. return address

Chapter 5, Multiple Choice #5: Someone searching with a search engine would use the

- A. crawler
- B. query processor
- C. index
- D. anchor

Problem 2: HCI [3 marks]

A professor recently gave a midterm exam that went very poorly. Students complained that they couldn't understand what the exam was asking for.

At the top of the exam there was a blank for students to fill in the names of their lab TAs. Only half the students were able to remember their TAs' names and fill it in! Keeping in mind the reason that commands are easier to remember in a GUI than in a command-line interface, suggest a design that would work better. Be sure to explain the connection between your design and the difference between GUIs and command-line interfaces!

Problem 3: Networks [3 marks]

In the party protocol for communication over broadcast networks, when two or more messages "collide" on the network, everyone picks their own random amount of time to wait before trying to send their message again. Why pick a random time rather than just setting the time to the same value for everyone?

Problem 4: Labs (ImageViewer) [3 marks]

Answer any **one** of the following questions from the ImageViewer lab in the space below.

1. `var number = Math.floor(Math.random()*10);`
generates a random number between 0 and 9. How would you generate a random number between 11 and 35?

2. There are unknown values stored in `a[0]`, `a[1]`, `a[2]`. How would you set `a[0]` to `a[2]`, `a[1]` to `a[0]`, and `a[2]` to `a[1]` without losing any of the values?

3. Given:

```
function sum(a, b){
    result = a + diff(b, a);
    return result;
}
function diff(x, y){
    return x - y;
}
```

What is returned on this function call: `sum(3, -5)`?

4. Why do you think dividing up code into functions is a good idea? Give at least 2 reasons.

Problem 5: JavaScript, HTML, and Algorithms [17 marks]

Consider the following web page code for a guessing game:

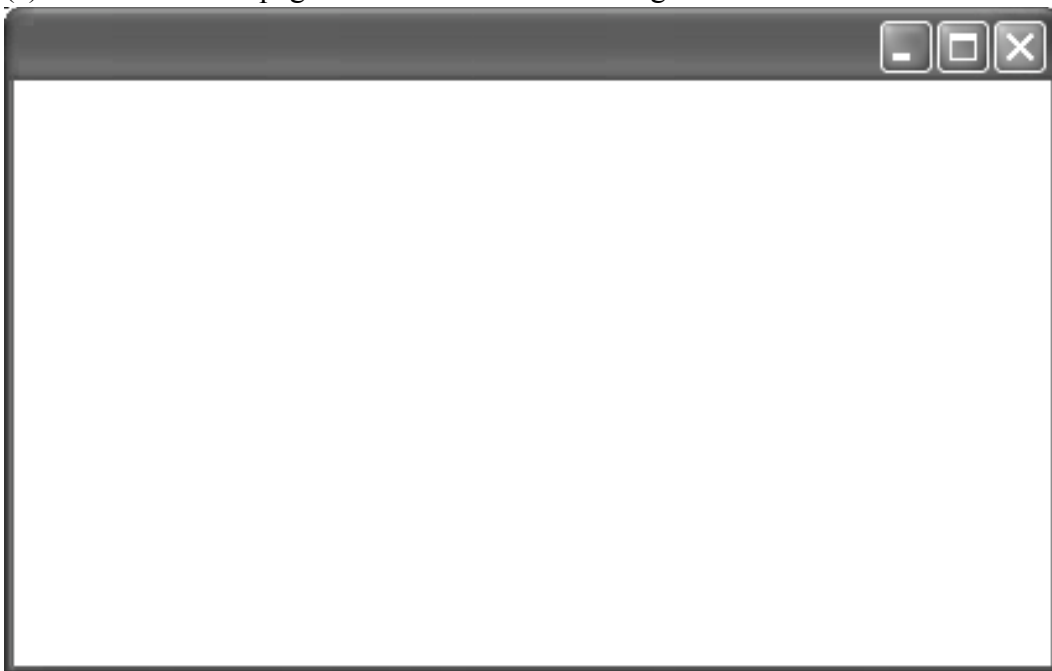
```
1 <html><head><title>Guessing Game</title>
2 <script language="JavaScript1.2">
3 var low = 1;    // The low end of the range the user's number is in.
4 var high = 31; // The high end of the range the user's # is in.
5 var guess = (low + high) / 2; // Our next guess at the user's #
6 var guesses = 1;           // The number of guesses so far
7
8 // Put the supplied text onto the page in the "gameBody" area.
9 function writeGame(text)
10 {
11     // Replaces the "div" element below with the given text.
12     document.getElementById("gameBody").innerHTML = text;
13 }
14
15 // Take one step in the game, showing appropriate text to the user.
16 // This game uses binary search to find the user's number.
17 function gameStep()
18 {
19     // If the range is down to one number, this guess is correct!
20     if (low == high)
21     {
22         result = "Your number is: " + low + "<br>";
23
24
25
26         result = result + "I found it!";
27
28
29
30         writeGame(result);
31     }
32     // Otherwise, make a new guess.
33     else
34     {
35         guess = (low + high) / 2;
36         guesses = guesses + 1;
37         writeGame("Is your number " + guess + "?");
38     }
39 }
40 </script></head>
41
42 <body>
43 <h1>Guessing Game</h1>
44 Pick a number between 1 and 31 and answer the question below.
45 <p>
46
47 <div id="gameBody">Is your number 16?</div>
48
49 <form name="guessGame" method="get" action="">
50
51 <!-- Too big!  Move the high end down below the guess. -->
52 <input type="button" value="No, it's smaller."
53     onclick="high = guess - 1; gameStep();">
54 <!-- We got it!  Set low and high to guess. -->
```

```

55 <input type="button" value="Yes!"
56     onclick="low = guess; high = guess; gameStep();">
57
58 <!-- Too small! Move the low end up above the guess. -->
59 <input type="button" value="No, it's bigger."
60     onclick="low = guess + 1; gameStep();">
61 </form>
62 </body></html>

```

- (a) In the code above, circle and clearly label one example each of: a variable, an assignment statement, an HTML comment, and an HTML header tag.
- (b) Which function is called when the user clicks the “Yes!” button?
- (c) Sketch what the page would look like after the game ends if the user’s number was 9.



- (d) What’s the largest number of guesses the computer could possibly use with this algorithm? (Hint: note that the code uses binary search and the user’s number must be between 1 and 31!)
- (e) How many **more** guesses might the computer have to use if the user was allowed to pick a number between 1 and 127 instead?
- (f) Add emphasis to the computer’s message on line 22. Use any reasonable HTML tag to add the emphasis. (Cross out the existing line and replace it with your own.)
- (g) Change line 26 so that the user sees text like: “That took 3 guesses.” rather than “I found it!” The number displayed should be the number of guesses the computer used.

Problem 6: Computers and Visual Art [10 marks]

Answer each of the following.

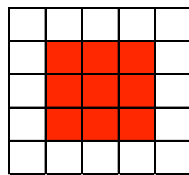
1. [4 points] List two limitations faced by computer artists in the 1960s and explain.
2. [4 points] What impact did the invention of raster graphics have on computer art? List two and explain.
3. [2 points] What is a fractal? Why are computers useful for representing fractals?
4. [2 points] Name two computer artists

Problem 7: Painting in JavaScript [5 marks]

(a) If you increase the x coordinate of a point in JavaScript painting by 10, does the point move left or right?

(b) If you increase the y coordinate by 10, does the point move up or down?

(c) You are using JavaScript to paint on a canvas that's 5 pixels wide and 5 pixels tall. Write in parameters for function `fillRect` in the code below so that it creates a rectangle that almost fills the canvas, leaving a 1 pixel wide border, like this:

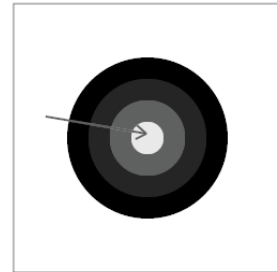
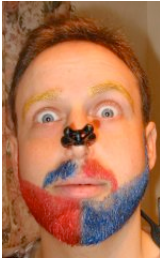


```
canvas.setColor("red");  
canvas.fillRect(      ,      ,      ,      );  
canvas.paint();
```

Note: `fillRect(x, y, w, h)` fills a rectangle at `(x, y)` with width `w` and height `h`.

Problem 8: Paint Programs and Number Representation [18 marks]

(a) For each of the following images, indicate whether it would be better to use GIF or JPEG compression to compress it:



(b) Why did the use of vector graphics in early computer displays (like pen plotters) limit artists' use of real photographic material?

(c) Use the term “pixel” to explain what a raster graphics display is.

(d) Circle one word on each of the following lines to describe the HTML color #E0E00E

circle one: **BRIGHT** **DARK**

circle one: **RED** **YELLOW** **GREEN** **BLUE-GREEN** **BLUE** **PURPLE**

(e) Fill in the table below so that each row contains the same value in all three systems:

Binary	Decimal	Hexadecimal
	4	
1010		
		11
11011100	220	
	161	A1

(f) Would the number 100000000000000000000000000000 be largest if it were a binary, decimal, or hexadecimal number?

EXTRA SPACE

(if you use this, CLEARLY indicate both the connection between this work and the problem it is for both here and where the problem is stated!)

Appendix A: Commonly used HTML tags and their meanings

Start Tag	End Tag	Meaning
<html>	</html>	HTML document; first and last tags in an HTML file
<title>	</title>	title bar text; describes page
<head>	</head>	preliminary material, e.g. title, at start of the page
<body>	</body>	the main part of the page
<p>	</p>	paragraph, can use align attribute
<hr>		line (horizontal rule), can use width and size attributes
<h1> ... <h8>	</h1> ... </h8>	headings, eight levels, use in order, can use align attribute
		bold
<i>	</i>	italic
		anchor reference, <i>fn</i> must be a pathname to a file, e.g. html or pdf file
		image source reference, <i>fn</i> must be pathname to image file (e.g. .jpg or .gif)
		ordered list
		unordered list
		list item
<table>	</table>	table, can use border attribute
<th>	</th>	table heading
<td>	</td>	table data
<input type="fn">		input field, <i>fn</i> must be an input type like button or radio; can use attributes like value (for button text) and onclick (for button behaviour when clicked)

Appendix B: Useful Tables for Number Conversion

decimal	binary	hexadecimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7

decimal	binary	hexadecimal
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Powers of 2:

27	26	25	24	23	22	21	20
128	64	32	16	8	4	2	1