Computing 101 Introduction to Computing
Summer 2001 - Solutions to Homework Assignment #1

Department of Computing Sciences
University of Alberta

Chapter 2:

Exercise 1a. Set the value of *area* to $\frac{1}{2}b \cdot h$

Exercise 1c. Set the va;ue of *FlyingTime* to $M/AveSpeed$

Exercise 4. Algorithm:

Step 1. Get the values of L, W and C
Step 2. Set the value of *Area* to $L \cdot W/9$ to get the area in square yards
Step 3. Set the value of *TotalCost* to $1.06 \cdot Area \cdot C$
Step 4. Print the value of *TotalCost*

Exercise 9. Algorithm:

Step 1. Repeat Steps 2 - 5 until $TotalCost < 1000$
Step 2. Get a value for L, W and C
Step 3. Set the value of *Area* to $L \cdot W/9$
Step 4. Set the value of *Total Cost* to $1.06 \cdot Area \cdot C$
Step 5. Print the value of *TotalCost*

Exercise 11. Algorithm:

Step 1. Repeat Steps 2 - 17 until $Response = 'No'$
Step 2. Get values of *Hours* and *Rate*
Step 3. If $Hours > 54$ then
Step 4. $DoubleTime = Hours - 54$
Step 5. $TimeAndHalf = 14$
Step 6. $Regular = 40$
Step 7. else if $Hours > 40$ then
Step 8. $DoubleTime = 0$
Step 9. $TimeAndHalf = Hours - 40$
Step 10. $Regular = 40$
Step 11. else $DoubleTime = 0$
Step 12. $TimeAndHalf = 0$
Step 13. $Regular = Hours$
Step 14. $GrossPay = Rate \cdot Hours + 1.5 \cdot Rate \cdot TimeAndHalf + 2.0 \cdot Rate \cdot DoubleTime$
Step 15. Print the value of *GrossPay*
Step 16. Print the message "Do you wish to do another computation?"
Step 17. Get the value of *Response*

Exercise 14. Assume that *FindLargest* is now a primitive operation in our pseudocode and use it to repeatedly remove the largest element from the list until we reach the median.

Algorithm:

Step 1. Get the value of N, and the values $L_1, L_2, \ldots, L_N$ in the list

Step 2. If N is even, then set $M = N/2$

Step 3. else let $M = (N + 1)/2$

Step 4. Repeat Steps 5 to 10 until $N < M$

Step 5. Use *FindLargest* to find the location *Loc* of the largest element in the list $L_1, L_2, \ldots, L_N$

Step 6. Exchange L_{Loc} and L_N as follows

Step 7. $Temp = L_N$

Step 8. $L_N = L_{Loc}$

Step 9. $L_{Loc} = Temp$

Step 10. Set N to $N - 1$ and shorten the list

Step 11. Print the message "The median is:"

Step 12. Print the value of L_M

Step 13. Stop

Chapter 3.

Exercise 4. *Legit* = 6

1 2 3 4 5 6 6 6 6

Number of Copies: 16

Exercise 5. *Legit* = 6

1 6 2 5 3 4 5 0 6

Number of Copies: 3

Exercise 6. Once item N has been copied one cell left, it need not be copied again. Similarly, once item $N - 1$ has been copied one cell left, it need not be copied again. The value of *Legit* shows how many cells from the right have been copied.

Step 11 of the algorithm can be changed to

"Repeat Steps 12 and 13 until *Right* > *Legit* +1"

Only 13 copies are done

Exercise 14b. The list after each exchange in *BubbleSort* is shown below.

12	3	6	8	2	5	7
3	**12**	6	8	2	5	7
3	**6**	**12**	8	2	5	7
3	6	**8**	**12**	2	5	7
3	6	8	**2**	**12**	5	7
3	6	8	2	**5**	**12**	7
3	6	8	2	5	**7**	**12**
3	6	**2**	**8**	5	7	12
3	6	2	**5**	**8**	7	12
3	6	2	5	**7**	**8**	12
3	**2**	**6**	5	7	8	12
3	2	**5**	**6**	7	8	12
2	**3**	5	6	7	8	12

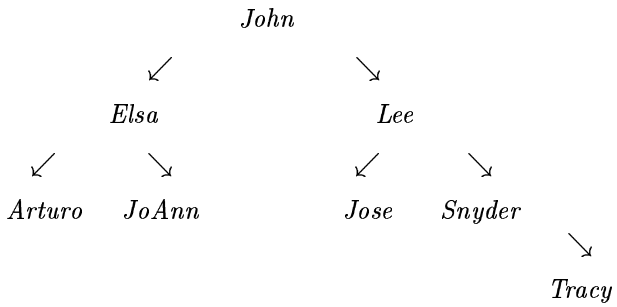
Bubblesort required more exchanges than selection sort on the input above.

Exercise 17c. The names compared with *Emile* are

John Elsa JoAnn

and there are only 3 comparisons made to determine that *Emile* is not in the list.

Exercise 21. The binary search tree is shown below.



Worst Case: 4 comparisons
Would occur if searching for Tracy