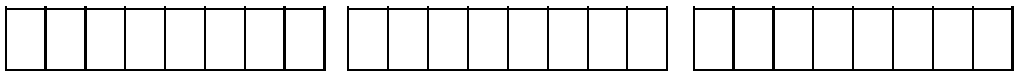
Lab1 - Data Representation and Pseudocode

Data Representation

1. Consult the table of ASCII character set in our handout. Show the binary (base two) representation (in contiguous memory) of the word "dog" (option: you can use "cat" instead).



For all problems below, show your work.

2. Convert the following binary string to hexadecimal notation.

   `1 0 1 1 0 0 1 1  1 0 0 1 1 1 0 0  0 0 1 0 1 1 1 1`

3. Convert the following hexadecimal number to binary.

   `b 6 3 e`

4. Convert the following binary string to base ten.

   `1 0 1 1 0 0 1 1`

5. Use Horner's algorithm to convert the following hexadecimal number to base ten.

   `3 d a 7`

6. Use the remainder method to convert the following base ten number to hexadecimal.

   `2467`

Introduction to Algorithms and Pseudocode

7. The following proposed algorithm is very similar to the one presented previously for finding the largest element in an array, but with one important difference. Trace this algorithm with the given input array. What happens ? What do you conclude about this proposed algorithm ?

   Input: An integer `n`, and an array `A` containing `n` integers.

   Output: The largest integer in the array.

   Method:

```
t = A[0]              // A[0] is our first candidate for the largest.
i = 1                 // i is our variable index.
while ( i < n ) {     // Start a loop.
   if ( t < A[i] ) {
      t = A[i]
      i = i + 1
   }
}                     // end while loop
write t               // Output the answer
```

22	37	11	17	30
0	1	2	3	4

8. Refer to the (correct) algorithm in the handout to find the largest element of the array. Modify that algorithm to reverse the elements in the array.

   Hint: Use a variable named x to (temporarily) hold the data value you are going replace.

   Hint: We number the positions in an array starting with zero. If there are n data item in an array, starting our arrays with position zero implies that the last data item is in position $n - 1$. For example, refer to the diagram in problem 7. There are five data items, in positions: 0, 1, 2, 3, and 4. The last data item is in position four.

   Hint: Use an index variable i to step through the positions in the array from left to right. In each position, exchange the data item in that position with the corresponding position counting from the right end of the array. Limit the value of i so that you step through only the first half of the array. If you step through the entire array, you will exchange each element twice, and end up with the original order of the data items in the array.