COS 126 **General Computer Science** **Fall 2005**

Exam 1 Solutions

1. **Number Systems (4 points)**
   a) `FEC9`
   b) `-311`
   c) `26E` (Note that 622 is 2×311.)

2. **Java Programming, Arrays, Standard Input (5 points)**
   a) This program read in a list of presidents from Standard Input and stores the names in parallel arrays. The number of each president determines the index in each name array. It also accepts a command line input requesting the nth president and prints out that president's name.
   b) `java Presidents 3 < US_Presidents.txt`
   c) `Thomas Jefferson`

3. **Adventures in Java Programming and Debugging (8 points)**
   a)

line number	Correct code
3	`int N = Integer.parseInt(args[0]);`
7	`if (N == 0) System.out.println(0);`
9	`for (int i = N; i != 0; i++)`

   Declaring `int i` on line 9 fixes all the compiler errors on lines 9 and 10.

   b)

line number	Correct code
9	`for (int i = sum - 1; i > 0; i--)`

   There are other acceptable answers.

   c)

N	sum	i	output
-2	2	1	
	3	0	-3

4. **Recursion (8 points)**
   (a) `foo(2)` returns `true`.
   (b) `foo(3)` returns `false`.
   (c) `foo(4)` returns `true`.
   (d) `foo(5)` returns `false`.
   (e) `foo(-1)` results in a StackOverflow error.

(f) `foo(5000)` returns true.

(g) `foo(5000)` generates `2500` calls to `bar`.

(h) In general, `foo(n)` determines whether `n` is even.

5. **Analysis of Algorithms (4 points)**

a) N^2

The outer loop is traversed N times. The inner loop is traversed 0 times, then 1 time, then 2 times, then 3 times . . . So, the number of operations is $0+1+2+...+N$ times, and the order of growth is N^2.

b) $N \log N$ just like quicksort.

c) constant

d) $N \log N$

The recursion tree for this method has $\log N + 2$ levels. For each level l, where $l \leq \log N$, there are 2^l calls to the recursive function. (Level 0 is the first call using the original N; level $\log N + 1$ contains the base case manypeak(0, ...).) Each call executes a loop that iterates $N/(2^l)$ times. Therefore, all the work done by all the looping on one level is N, for $N \log N$ overall order of growth.

6. **Arrays, Functions, Analysis of Algorithms (8 points)**

a) `mystery1(a, 5)` returns `true`.

b) Fill in the trace table to show that `mystery2(a, 5)` returns the same thing.

target	low	high	mid	return value
5	0	6	3	
		2	1	true

c) `mystery1(a, 20)` and `mystery2(a, 20)` both return `false` .

d) These methods check whether `target` is an element in the array.

e) `mystery1(a, 32)` makes 14 comparisons with the target. (two comparisons each pass through the `for` loop)

f) `mystery2(a, 32)` makes 6 comparisons with the target. (two comparisons each pass through the `while` loop)

g) `mystery2()`

h) `mystery1()` does a sequential search through the array elements, so it has possibly `N` passes through the loop.

`mystery2()` halves the search area each pass, so worst case, it makes $\log N$ passes through the loop.

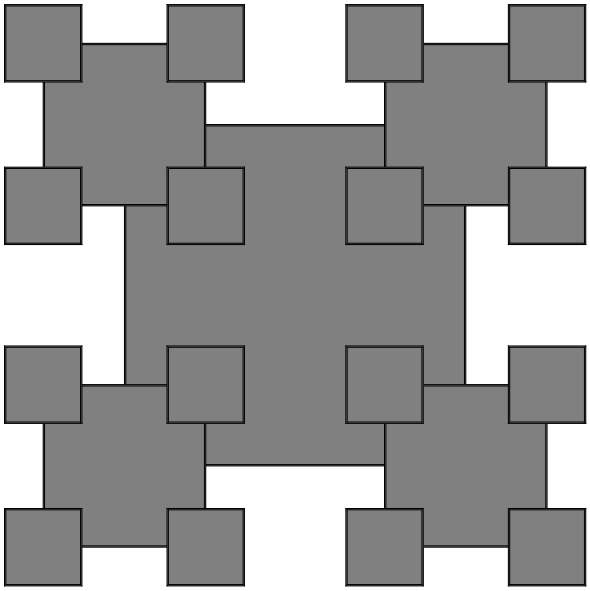
7. **Recursive Graphics (5 points)**

i) C, F

ii) none

iii) A

iv) Draw the picture that will result from ordering E when `n = 2`.



8. **TOY (6 points)**

a) Trace through the program for the following data.

```
// data
50: 0003
51: 0007
52: 0005
53: 0006
```

R[1]	R[2]	R[3]	R[4]	R[A]	R[B]	R[F]	output
0001	FFFE	0007		0051	0003	0013	
	0001	0005	0005	0052	0002		
			0006	0053	0001		
					0000		0005

b) Program will output `0002` (the smallest element in the array).

c) `AE` (hex) or `174` (decimal).
The array can use locations `51` thru `FE`. (hex)