COS 126	General Computer Science	Fall 2005

Exam 1

This test has 9 questions worth a total of 50 points. You have 120 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet, one side only, handwritten by you. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

__________________________-

Signature

Problem	Score	Possible
0		2
1		4
2		5
3		8
4		8
5		4
6		8
7		5
8		6
Total		50

Name:

NetID:

Preceptor: Frances Forrester

Haakon Andrea

Donna

Curtis Eric

The TOY instruction cheatsheet is on the last page of the exam.

There is no list of StdDraw methods, because none of the questions require you to write graphics code.

Feel free to tear out the last page in order to use the TOY reference more easily and to use the blank side for scratch paper.

0. **Miscellaneous. (2 points) (really)**

   (a) Write your name and Princeton NetID in the space provided on the front of the exam, and circle the name of your preceptor.

   (b) *Copy* and *sign* the honor code on the front of the exam.

1. **Number Systems (4 points)**

   a) Give the hexadecimal representation of the binary integer `1111111011001001`. Circle your answer.

   b) Give the decimal representation of the 16 bit 2's complement binary number `1111111011001001`. Circle your answer.

   c) Convert the decimal number 622 to hexadecimal. Circle your answer.

This page intentionally left blank.

2. **Java Programming, Arrays, Standard Input (5 points)**

You compile the following program:

```
public class Presidents {
   public static void main(String[] args) {

      //
      int N = StdIn.readInt();
      String[] firstNames = new String[N];
      String[] lastNames = new String[N];

      //
      for (int i = 0; i < N; i++){
         String last = StdIn.readString();
         String first = StdIn.readString();
         int number = StdIn.readInt();
         firstNames[number-1] = first;
         lastNames[number-1] = last;
      }

      //
      int x = Integer.parseInt(args[0]);
      if (x < 1 || x > N) System.out.println("Invalid request");
      else System.out.println(firstNames[x-1] + " " + lastNames[x-1]);
   }
}
```

You have the following data file, US_Presidents.txt, containing the names of the first 10 U.S. presidents in alphabetical order.

```
10
Adams John 2
Adams John 6
Harrison William 9
Jackson Andrew 7
Jefferson Thomas 3
Madison James 4
Monroe James 5
Tyler John 10
VanBuren Martin 8
Washington George 1
```

2. continued

   a) In general, what does this program do?

   b) What would you type to run this program using the given input file and a command line input of 3?

   c) What is the output that results from a command line input of 3?

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

The following program is supposed to take command line input N and compute the sum from 1 to $|N|$ (absolute value of N). It should print the sum if N is positive and print -sum if N is negative. (e.g., an input of 2 should output 3 and an input of -2 should output -3)

```
 1   public class BuggyCode {
 2      public static void main(String [] args)  {
 3         int N = args[0];
 4         int sum;
 5         if (N < 0) sum = -N;
 6         else sum = N;
 7         if (N = 0) System.out.println(0);
 8         else {
 9            for (i = N; i != 0; i++)
10               sum = sum + i;
11            if (N < 0) System.out.println(-sum);
12            else System.out.println(sum);
13         }
14      }
15   }
19
```

a) Running `javac BuggyCode.java` yields the following compilation time errors.

```
BuggyCode.java:3: incompatible types
BuggyCode.java:7: incompatible types
BuggyCode.java:9: cannot resolve symbol : variable i
BuggyCode.java:9: cannot resolve symbol : variable i
BuggyCode.java:9: cannot resolve symbol : variable i
BuggyCode.java:10: cannot resolve symbol : variable i
BuggyCode.java:10: incompatible types
```

How do you fix the compilation time bugs?

`line number     Correct code`

3. Continued

   b) Now that it compiles correctly, you run it with an input of 2 and you wait, and you wait, and you wait. Finally, you hit `Ctrl-C`. Which line is causing the trouble? Fix it so that the code now runs correctly.

      `line number     Correct code`

   c) Trace through your corrected code for an input of -2. Fill out the table below for each listed variable.

N	sum	i	output

4. **Recursion (8 points)**

Consider the following mutually recursive methods.

```
public static boolean foo(int x){
   if (x == 0) return true;
   else return bar(x-1);
}

public static boolean bar(int x){
   if (x == 0) return false;
   else return foo(x-1);
}
```

(a) What will `foo(2)` return?

(b) What will `foo(3)` return?

(c) What will `foo(4)` return?

(d) What will `foo(5)` return?

(e) What does `foo(-1)` do?

(f) What will `foo(5000)` return?

(g) How many calls to `bar` will `foo(5000)` generate?

(h) In general, what does `foo(n)` determine?

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

   For each of the four Java methods (i.e., functions) below, circle the order of growth of the average running time of the method. (Assume operations like + and * take a constant amount of time.) Don't worry about the numbers getting too big to fit in an int.

a)
```
public static void initTriangle(int[][] A){
// A is a square matrix (NxN).
   int N = A.length;  // For a 2D array, array.length is number of rows
   for (int i = 0; i < N; i++)
      for (int j = 0; j < i; j++)
          A[j][i] = -A[i][j];
}
```

constant $\quad \log N \quad N \quad N\log N \quad N^2 \quad 2^N$

b)
```
public static double median (double[] A) {
   int N = A.length;
   quicksort(A);  //sort the elements of A using quicksort
   if ((N % 2) == 1) return A[N/2];
   else return (A[N/2 - 1] + A[N/2]) / 2.0;
}
```

constant $\quad \log N \quad N \quad N\log N \quad N^2 \quad 2^N$

c)
```
public static double getMedian (double[] A) {
// given an ordered array of length N, return the median
   int N = A.length;
   if ((N % 2) == 1) return A[N/2];
   else return (A[N/2 - 1] + A[N/2]) / 2.0;
}
```

constant $\quad \log N \quad N \quad N\log N \quad N^2 \quad 2^N$

d)
```
public static void manypeak (int N, double size) {
   if (N > 0) manypeak(N/2, size/2);

   for (int i = 1; i <=N; i++) {
      drawpeak(size);     //method drawpeak() takes constant time
   }

   if (N>0) manypeak(N/2, size/2);
}
```

constant $\quad \log N \quad N \quad N\log N \quad N^2 \quad 2^N$

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

The following two methods do the same job. They each take an ORDERED array of `ints` and a target number as arguments.

```
public static boolean mystery1(int[] array, int target) {
   for (int i = 0; i < array.length; i++) {
      if (array[i] == target) return true;
      else if (array[i] > target) return false;
   }
   return false;
}

public static boolean mystery2(int[] array, int target) {
   int low = 0;
   int high = array.length - 1;
   while (low <= high) {
      int mid = (low + high) / 2;
      if (array[mid] == target) return true;
      else if (array[mid] < target) low = mid + 1;
      else high = mid - 1;
   }
   return false;
}
```

Use this array to answer the following questions:

```
int[] a =  { 2, 5, 11, 14, 15, 27, 31};
```

a) What does `mystery1(a, 5)` return?

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

target	low	hi	mid	return value

6. Continued

c) What do `mystery1(a, 20)` and `mystery2(a, 20)` return?

d) In general, what do these methods do?

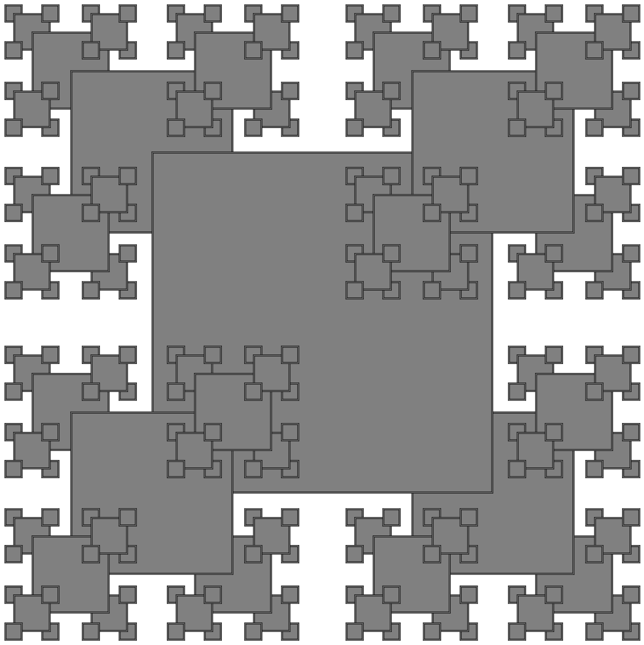
e) How many comparisons with the target will `mystery1(a, 32)` make?

f) How many comparisons with the target will `mystery2(a, 32)` make?

g) Which one would you use for an array of a million elements?

h) Why?

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



Suppose that the method `drawShadedSquare()` draws a gray square with a black outline. The square is centered at `(x,y)` and has side length `size`. Consider the following recursive function.

```
   public static void recur(int n, double x, double y, double size){
1    drawShadedSquare(x, y, size);
2    recur(n-1, x + size/2., y + size/2., size/2.1);
3    recur(n-1, x + size/2., y - size/2., size/2.1);
4    recur(n-1, x - size/2., y - size/2., size/2.1);
5    recur(n-1, x - size/2., y + size/2., size/2.1);
6    if (n <= 0) return;
   }
```

The method call `recur(n, 256.0, 256.0, 256.0)` is supposed to produce the recursive pattern above. All 6 numbered statements are correct, but they are not necessarily in the correct order.

Here are some proposed orderings:

A) `1 2 3 4 5 6`

B) `6 1 3 5 4 2`

C) `6 3 5 1 4 2`

D) `6 2 4 1 3 5`

E) `1 6 3 4 5 2`

F) `6 5 3 1 2 4`

7. Continued

i) If `n = 5`, which, if any, of the above orderings of statements could produce the picture shown above?

List the letters of all that apply or write `none`.

ii) If `n = 6`, which, if any, of the above orderings of statements could produce the picture shown above?

List the letters of all that apply or write `none`.

iii) If `n = 4`, which, if any, of the listed orderings of statements could produce a StackOverflow Error?

List the letters of all that apply or write `none`.

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

8. **TOY (6 points)**

```
// main program
10: 7A51   R[A] <- 0051
11: 8B50   R[B] <- mem[50]
12: FF20   R[F] <- pc; goto 20
13: 0000   halt
////////////////////////////////////////
// Function: TOY_Mystery
20: 7101   R[1] <- 0001
21: A30A   R[3] <- mem[R[A]]
22: 2BB1   R[B] <- R[B] - R[1]
23: CB2A   if (R[B] == 0) goto 2A
24: 1AA1   R[A] <- R[A] + R[1]
25: A40A   R[4] <- mem[R[A]]
26: 2243   R[2] <- R[4] - R[3]
27: D222   if (R[2] > 0) goto 22
28: 1340   R[3] <- R[4]
29: C022   goto 22
2A: 93FF   write R[3]
2B: EF00   goto R[F]
```

Your main program starts at address 10.

There is a function which begins at address 20.

There is an array in memory starting at address 51.

Memory location 50 contains the size of the array.

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

8. Continued

b) What will the program output using the following data? If you have figured out what the function does, you do not need to make a full trace, but here is a table to use, just in case you need to do a few more iterations.

```
// data
50: 000F
51: 0007
52: 0005
53: 0006
54: 0027
55: 00A5
56: 0060
57: 001B
58: 0002
59: 00A6
5A: 0038
5B: 003C
5C: 0012
5F: 0021
5E: 000A
5F: 0014
```

R[1]	R[2]	R[3]	R[4]	R[A]	R[B]	R[F]	output

c) If you store the array sequentially in the TOY machine starting at memory address `51`, and your entire program, including functions, is stored in `00` through `4F`, what is the maximum number of elements that the array can safely have? Your answer may be in decimal or hex, but you must indicate what base you are using. Circle your answer.

This page intentionally left blank.

TOY REFERENCE CARD

```
  INSTRUCTION FORMATS
           | . . . . | . . . . | . . . . | . . . . |
Format 1:  | opcode  |    d    |    s    |    t    | (0-6, A-B)
Format 2:  | opcode  |    d    |       addr        | (7-9, C-F)
```

```
  ARITHMETIC and LOGICAL operations
1:  add          R[d] <- R[s] + R[t]
2:  subtract     R[d] <- R[s] - R[t]
3:  and          R[d] <- R[s] & R[t]
4:  xor          R[d] <- R[s] R̂[t]
5:  shift left   R[d] <- R[s] << R[t]
6:  shift right  R[d] <- R[s] >> R[t]
```

```
  TRANSFER between registers and memory
7:  load address    R[d] <- addr
8:  load            R[d] <- mem[addr]
9:  store           mem[addr] <- R[d]
A:  load indirect   R[d] <- mem[R[t]]
B:  store indirect  mem[R[t]] <- R[d]
```

```
  CONTROL
0:  halt             halt
C:  branch zero      if (R[d] == 0) pc <- addr
D:  branch positive  if (R[d] > 0) pc <- addr
E:  jump register    pc <- R[d]
F:  jump and link    R[d] <- pc; pc <- addr
```

Register 0 always reads 0.
Loads from mem[FF] come from stdin.
Stores to mem[FF] go to stdout.