COS 126 **General Computer Science** **Spring 2014**

Written Exam 1

This exam is closed book, except that you are allowed to use a one-page single-sided cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided.

Print your name, login ID, lecture number and precept number on this page (now), and write out and sign the Honor Code pledge before turning in this paper. It is a violation of the Honor Code to discuss this exam until everyone in the class has taken the exam. You have 50 minutes to complete the test.

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."

Pledge: _____

_____

Signature: _____

Name: _____

NetID: _____

Registered Lecture: _____

Precept: _____

L01	10:00 TTh	
L02	11:00 TTh	
P01	12:30 TTh	David Pritchard
P01A	12:30 TTh	Donna Gabai
P01B	12:30 TTh	Aleksey Boyko
P02	1:30 TTh	Borislav Hristov
P02A	1:30 TTh	Terry Yannan Wang
P02B	1:30 TTh	Aleksey Boyko
P02C	1:30 TTh	Nanxi Kang
P02D	1:30 TTh	Xinyi Fan
P03	2:30 TTh	Borislav Hristov
P03A	2:30 TTh	Bebe Shi
P04	3:30 TTh	Kevin Lee
P04A	3:30 TTh	Victor Shaoqing Yang
P05	7:30 TTh	Kevin Lee
P06	10:00 WF	Mojgan Ghasemi
P07	11:00 WF	Mojgan Ghasemi
P08	12:30 WF	Donna Gabai
P08A	12:30 WF	Maia Ginsburg
P09	1:30 WF	David Pritchard
P09A	1:30 WF	Maia Ginsburg
P09B	1:30 WF	Judi Israel
P10	2:30 WF	Judi Israel

Problem	**Value**	**Score**
0*	1	
1	11	
2	10	
3	8	
4	9	
5	8	
6	8	
7	8	
8	7	
Total	**70**	

* Question 0: Did you show up to the right room at the right time?

1 Java Expressions (11 points)

For each of the Java expressions below, write down the type of the expression and its value. If the expression causes a syntax or run-time error, write an X in both boxes. One of the entries is filled in for you.
For the bottom one, use the following array:

```
int[] a = {1, 2, 3, 4, 5};
```

	Type	Value
`"0" + 1/2 + 3*4`	String	"0012"
`Integer.parseInt("1e100")`	X	X
`Double.parseDouble("1" + "2")`	double	12.0
`true != true != true`	boolean	true
`((int) Math.PI) + ((int) 1.7)`	int	4
`a[a[a.length/2]]`	int	4

2 Number Systems, Bitwise Operators (10 points)

For this problem, we ask you to perform several calculations on hexadecimal and binary numbers. Unless noted otherwise, all given numbers are in **16-bit twos-complement hexadecimal** representation. **All** answers should also be in **16-bit twos-complement hexadecimal**.

- What is `CAFE & FACE`?

  CACE

- Which of the following three numbers is largest? Circle one of the three choices.

  `2014` (`7ABC`) the result of adding `2014` and `7ABC`

- What is `0123 >> 0001`?

  0091

- How do you express −4 (decimal) in 16-bit twos-complement hexadecimal representation?

  FFFC

- The average COS 126 student has submitted 680 lines of code so far this semester. What is decimal 680, expressed in 16-bit twos-complement hexadecimal representation?

  02A8

3 Input/Output, Pipes, Redirection (8 points)

Consider the following program:

```
1  public class DoSomething {
2      public static void main(String[] args) {
3          int first = StdIn.readInt();
4          int remember = first;
5          while (!StdIn.isEmpty()) {
6              int next = StdIn.readInt();
7              StdOut.print(remember + next + " "); // a number then a space
8              remember = next;
9          }
10         StdOut.println(remember + first); // a number
11     }
12 }
```

Suppose that `input.txt` contains the text: `1 2 3 4`

(a) What is the output of this command?

`java-introcs DoSomething < input.txt`

3 5 7 5

(b) What is the output of this command?

`java-introcs DoSomething < input.txt | java-introcs DoSomething`

8 12 12 8

(c) What are the contents of the text file `mystery.txt`, if the output of

`java-introcs DoSomething < mystery.txt`

is the single number `126`?

63

4 Debugging (9 points)

Willy Woodrow is trying to write a program that will repeatedly print any string any number of times. For example, he'd like `java-introcs Repeat Hello 3` to print:

```
HelloHelloHello
```

After spending some quality time with DrJava, Willy has arrived at the following buggy code to perform this task.

```
1  // execution: java Repeat sometext N
2  // N should be a positive integer
3  public class Repeat {
4      public static void main(String[] args) {
5          String text = args[0];
6          int repeats = Integer.parseInt(args[0]);
7          int i = 0.0;
8          while (i < repeats) {
9              i =+ 1;
10             StdOut.print(text);
11         }
12     }
13 }
```

(a) When Willy tries to compile his program, he encounters a "possible loss of precision" error. What line is causing this error? What is the correct line of code?

Line number: 7

Correct code: int i=0; OR double i=0.0;

(b) After fixing the previous bug, Willy is able to compile the program. He then runs

`java-introcs Repeat Hello 3`

but this causes a `NumberFormatException`. What line is causing this exception, and what character must be changed in order to fix it?

Line number: 6

Character that must be changed: 0 What it must be changed to: 1

(c) After fixing this second bug, Willy runs `java-introcs Repeat Hello 3` again. This time, the program runs forever; it keeps printing `Hello` over and over. What two characters (on the same line) must be swapped in order for this final bug to be fixed?

Line number: 9

Characters that must be swapped: = and +

TOY Reference Card *Use this for the next problem on the facing page.*

```
              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] ^  R[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.
pc starts at 10

16-bit registers
16-bit memory locations
 8-bit program counter
```

5 TOY (8 points)

(a) Write a single TOY command that always prints `0000` to standard output.

90FF

For parts (b), (c) and (d), assume that R[1] CONTAINS THE HEX VALUE 0001.

(b) Write a single TOY command that will put the hex value `FFFF` (negative one) in R[2].

2201

(c) Of the four commands below, two of them are equivalent to each other. Which ones are they? (Circle exactly two commands.)

`1ABB` `1BAA` `5AB1` `5B1A`

(d) Write a single TOY command that makes `R[6]` contain `0001` when `R[5]` is odd, and makes `R[6]` contain `0000` when `R[5]` is even.

3651 OR 3615

6 Scope, Pass by Value (8 points)

In this problem, you will analyze the program below:

```
public class Calls {
    public static void snap(int x) {
       x = 0;
    }

    public static void crackle(int[] a) {
       for (int i=0; i < a.length; i++)
          a[i] = a[i] + 1;
    }

    public static void pop(int[] arr) {
       arr = new int[arr.length];
    }

    public static void main(String[] args) {
       int x = 126;
       snap(x);
       System.out.println(x);

       int[] a = {1, 2, 6};
       crackle(a);
       for (int i=0; i < a.length; i++)  System.out.print(a[i] + " ");
       System.out.println();

       int[] arr = {20, 14};
       pop(arr);
       for (int i=0; i < arr.length; i++)  System.out.print(arr[i] + " ");
       System.out.println();
    }
}
```

What output is printed by this program?

- First line of output: **126**
- Second line of output: **2 3 7**
- Third line of output: **20 14**

7 Doubling Method, Orders of Growth (8 points)

We would like to write a program that reads any number of integers from standard input, and determines if any two of them are the same, or if they are all different. This is like `Distinct` from precept, but using `StdIn`. Let N be the number of integers; assume the program takes N as a command-line argument.

You create a program to solve this problem, using nested loops. When you run your program on inputs of various sizes, you obtain the following chart of running times, where N is the number of inputs:

N	running time
10000	0.3 seconds
20000	1.1 seconds
30000	2.6 seconds
40000	4.5 seconds

(a) Using the doubling hypothesis method, find values of a and b, with b an **integer**, so that $a \times N^b$ is an approximate model for the running time of this program.
(Feel free to leave a in scientific notation or as a numeric expression. Round off to one significant digit or leave as a fraction.)

a=0.3/(10000)²=3E-9 sec. **b=2**

For a, you can solve from other data points, e.g. 4.5/(40000)² is correct

a : _____ b : _____

(b) Thinking some more, you discover a more efficient two-part algorithm for this problem. You run your improved program and get the following running times:

N	running time
10000000	0.5 seconds
20000000	1.0 seconds
30000000	1.6 seconds
40000000	2.1 seconds

Using the doubling hypothesis method, find values of a and b, with b an **integer**, so that $a \times N^b$ is an approximate model for the running time of this improved program.

a=0.5/10000000=5E-8 sec. **b=1** **Same as previous comment.**

a : _____ b : _____

(c) Assume that in part (b), your improved method is

- First sort all of the numbers using **MergeSort**.
- Then check to see if any *adjacent* numbers are the same.

What do you think is the true order of growth of the running time of this improved program? Circle one choice.

Logarithmic Linear Linearithmic Quadratic Exponential

8 Recursion (7 points)

Here is a program that prints a sequence of numbers; note the lines labelled A, B, and C.

```
public class NumberSequence {
   public static int recur(int count, int n) {
      int newCount = count;
      if (n >= 1) {
         newCount = recur(newCount, n-1);                    // line A: recur
         newCount++; StdOut.print(newCount + "" + n + " "); // line B: increment, print
         newCount = recur(newCount, n-1);                    // line C: recur
      }
      return newCount;
   }
   public static void main(String[] args) {
      recur(0, 3);
      StdOut.println();
   }
}
```

For parts (a) and (b), in choosing your answer, refer to these three outputs and their labels:

```
label   output
(i)     13 22 31 41 52 61 71
(ii)    31 22 13 14 25 16 17
(iii)   11 22 31 43 51 62 71
```

(a) Which of the three outputs is generated when we run this program? Circle one of the following choices.

(i) (ii) (iii)

(b) Suppose we swap the order of lines A and B in the code. Which of the three outputs is generated when we run the modified program?

(i) (ii) (iii)

(c) Suppose we instead rearrange the code so that line A comes first, then line C, then line B. Fill in the 7 blanks in the diagram below with pairs of numbers, indicating what the output of the program would look like.

11 **21** **32** **41** **51** **62** **73**

PRINT your name here: _____

This page intentionally left blank for scratch paper. Return it with your test.