This test has 8 questions worth a total of 45 points. You have 50 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet. 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.”

<table>
<thead>
<tr>
<th>Problem</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

Name: ______________________________

NetID: ______________________________

Signature: _______________________

P01  TTh 1:30  Jen
P01A TTh 1:30  Tim
P01B TTh 1:30  Donna
P01C TTh 1:30  Doug
P01D TTh 1:30  Casey
P02  TTh 2:30  Doug
P03  TTh 3:30  Jen
P04  TTh 7:30  Aleksey
P05  WF 10    Kevin
P06  WF 1:30  Donna
P06A WF 1:30  Chris
P06B WF 1:30  Kevin
0. Miscellaneous. (1 point)

(a) PRINT your name and Princeton NetID in the space provided on the front of the exam, and circle your precept number.

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

1. TOY Programming. (5 points) The following (partial) TOY program inputs a positive integer N from standard input, and then sends to standard output the first N powers of two starting with one (e.g., if N=4 the output is 1 2 4 8). Fill in the 5 missing instructions in the blanks provided below.

10: ____ read R[C] from stdin input N, store in R[C]
13: 92FF write R[2] to stdout output
14: ____ ... multiply R[2] by 2
16: ____ ... loop if not done
17: 0000 halt
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 \( R[d] \leftarrow R[s] + R[t] \)
2: subtract \( R[d] \leftarrow R[s] - R[t] \)
3: and \( R[d] \leftarrow R[s] \& R[t] \)
4: xor \( R[d] \leftarrow R[s] \oplus R[t] \)
5: shift left \( R[d] \leftarrow R[s] \ll R[t] \)
6: shift right \( R[d] \leftarrow R[s] \gg R[t] \)

TRANSFER between registers and memory
7: load address \( R[d] \leftarrow \text{addr} \)
8: load \( R[d] \leftarrow \text{mem[addr]} \)
9: store \( \text{mem[addr]} \leftarrow R[d] \)
A: load indirect \( R[d] \leftarrow \text{mem[R[t]]} \)
B: store indirect \( \text{mem[R[t]]} \leftarrow R[d] \)

CONTROL
0: halt \( \text{halt} \)
C: branch zero if \( (R[d] == 0) \) \( \text{pc} \leftarrow \text{addr} \)
D: branch positive if \( (R[d] > 0) \) \( \text{pc} \leftarrow \text{addr} \)
E: jump register \( \text{pc} \leftarrow R[d] \)
F: jump and link \( R[d] \leftarrow \text{pc}; \text{pc} \leftarrow \text{addr} \)

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

16-bit registers (using two's complement arithmetic)
16-bit memory locations
8-bit program counter
2. **Scope. (6 points)** What is the output of the following program? Circle your answer.

```
public class PassByValue {

    public static void f(int x, int[] y, int[] z) {
        x = 1;
        y[0] = 2;
        z = new int[5];
        z[0] = 555;
    }

    public static void main(String[] args) {
        int x = 111;
        int[] y = { 222, 333, 444, 555 };
        int[] z = { 666, 777, 888, 999 };

        f(x, y, z);

        System.out.println(x);
        System.out.println(y[0]);
        System.out.println(z[0]);
    }
}
```
3. **Number Systems. (8 points)**

(a) What is the decimal representation of the 16-bit two’s complement integer 1111 1111 1010 1000_2? Circle your answer.

(b) Write the decimal integer -65 in TOY (16-bit two’s complement integer, in hexadecimal). Circle your answer.

(c) How many lines of output does the following Java code fragment produce?

```java
for (int i = 0; i >= 0; i++) {
    System.out.println(i);
}
```

Circle the best answer.

0 2^{31} - 1 2^{31} 2^{32} - 1 2^{32} infinite

(d) What is the value of `answer` when the Java loop terminates?

```java
int answer = 0;
for (int i = 0; i < 256; i++) {
    answer = answer ^ i;
}
```

Circle the best answer.

0 1 255 256 NaN integer overflow
4. Java Expressions. (9 points)

(a) Assume that \( a, b, \) and \( c \) are variables of type `boolean`. Consider the following three conditions.

i. \((a \; || \; b) \; && \; c\)

ii. \((a \; && \; b) \; || \; c\)

iii. \!(a \; && \; b) \; && \; c\)

Which of the expressions above is (are) always true if \( a \) and \( b \) have different values (e.g., one is true and one is false) and \( c \) is true? Circle the expressions that evaluate to true.

(b) What is the type and value of each of the following expressions?

<table>
<thead>
<tr>
<th>Expression</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 * 0.2</td>
<td>(int)</td>
<td>(11 * 0.2)</td>
</tr>
<tr>
<td>(int) 11 * 0.2</td>
<td>(int)</td>
<td>(11 * 0.2)</td>
</tr>
<tr>
<td>11 * (int) 0.2</td>
<td>(int)</td>
<td>(11 * 0.2)</td>
</tr>
<tr>
<td>(int) (11 * 0.2)</td>
<td>(int)</td>
<td>(11 * 0.2)</td>
</tr>
</tbody>
</table>

(c) Assuming \( i \) is a 32-bit integer, consider the following three expressions.

i. \((i \; >> \; 16) \; & \; 15\)

ii. \((i \; & \; 0x000F0000) \; >> \; 16\)

iii. \((i \; & \; 0x000FF000) \; >> \; 16\)

Which of the expressions are equivalent? Circle the best answer.

i and ii

ii and iii

i and iii

i, ii, and iii

None
5. Arrays. (9 points) The following program prints three lines of output. What are the three lines? Circle your answer.

```java
public class Mystery {
    public static void main(String[] args) {
        // Part 1
        int[] a = {0, 1, 2, 3, 4, 5};
        for (int i = 0; i < 6; i++)
            System.out.print(a[i/2] + " ");
        System.out.println();

        // Part 2
        for (int i = 5; i >= 0; i--)
            System.out.print(5 - a[a[i]] + " ");
        System.out.println();

        // Part 3
        for (int i = 0; i < 3; i++) {
            int t = a[i];
            a[i] = a[5-i];
            a[5-i] = t;
            System.out.print(a[i] + " ");
        }
    }
}
```
6. **Recursive Methods. (3 points)** The function below is supposed to sum all the odd, or all the even, positive integers up to n, depending on whether n is odd or even.

```java
public static int f(int n) {
    if (n==0 || n==1) return n;
    return n + f(n-2);
}
```

Circle any of the following that are correct:

(a) base case written incorrectly, so function will get the wrong answer, or no answer, for some valid positive values of n
(b) reduction step written incorrectly, so function will get the wrong answer, or no answer, for some valid positive values of n
(c) function is tail-recursive and therefore will get the wrong answer, or no answer, for some valid positive values of n
(d) this function looks fine to me
(e) it doesn’t have problems a, b, or c, but does have the following problem, which will cause it to get the wrong answer, or no answer, for some valid positive values of n (write the problem down)
7. **Arrays and I/O. (4 points)** Consider the following program.

```java
public class Mystery {
    public static void main(String[] args) {
        int N = args.length;
        String[] a = new String[N * 2];
        for (int i = 0; i < N; i++) {
            a[i] = args[i];
            a[i + N] = args[N - i - 1];
        }
        for (int i = 0; i < a.length; i++)
            StdOut.print(a[i] + " ");
        StdOut.println();
    }
}
```

(a) What does this program print out when the following command is executed?

```
% java Mystery aaa bbb ccc
```

(b) What does this program print out when the following command is executed?

```
% java Mystery aaa bbb ccc | java Mystery xxx yyy
```