0. Miscellaneous.

Don’t forget to write your name, NetID, precept, and exam room.

1. Java basics.

<table>
<thead>
<tr>
<th>Java expression</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>1.0 * x + y / z</td>
<td>1.0</td>
</tr>
<tr>
<td>((x &gt; y)</td>
<td></td>
</tr>
<tr>
<td>Math.min(Math.max(x, y), y*y % z)</td>
<td>1</td>
</tr>
<tr>
<td>Math.sqrt(x - y - z)</td>
<td>NaN or Double.NaN</td>
</tr>
<tr>
<td>Double.parseDouble(x + &quot;2&quot; + y)</td>
<td>122.0</td>
</tr>
</tbody>
</table>

2. Properties of arrays and functions.

(a) T T F F F F
(b) T T F F T

3. Debugging and arrays.

D F G A C
4. Functions.

(a) CKDKEK

The letters C, D, and E can be permuted in any order. EKKGKH is an alternative solution, but poorer style.

```java
public static boolean oddParity(boolean x, boolean y, boolean z) {
    int count = 0;
    if (x) count++;
    if (y) count++;
    if (z) count++;
    return (count % 2) != 0;
}
```

(b) GEHFE

GEHFJ, GIHFE, GIHFJ are alternative solutions, but poorer style.

```java
public static boolean oddParity(boolean x, boolean y, boolean z) {
    if (x && y) return z;
    else if (x || y) return !z;
    else return z;
}
```

This part is trickier than it might appear. It is tempting to start with IAGB, but then you quickly get stuck.

5. Recursion.

(a)

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>f(n)</td>
<td>&quot;0&quot;</td>
<td>&quot;1&quot;</td>
<td>&quot;1201&quot;</td>
<td>&quot;1201311201&quot;</td>
</tr>
</tbody>
</table>

(b) T T T

- Let $L_n$ be the length of the return value of $f(n)$. Then, $L_n = L_{n-1}+1+L_{n-2}+L_{n-1}$, with $L_0 = L_1 = 1$. From above, $L_2 = 4$ and $L_3 = 10$. Thus, $L_4 = 10+1+4+10 = 25$ and $L_5 = 25+1+10+25 = 61$.
- Exchanging statements 2 and 3 can have an effect only when $n$ is 1. When $n$ is 1, the modified function still returns "1"—it just performs the unnecessary work of calling $f(0)$ before doing so.
- The function $f()$ has no side effects (such as printing to standard output). Thus $\text{first} = f(n-1)$ and $\text{third} = f(n-1)$ will always be equal, and the return value in the original function ($\text{first} + n + \text{second} + \text{third}$) will always equal the return value in the modified function ($\text{first} + n + \text{second} + \text{first}$).

7. TOY.

The TOY program computes the smallest power of 2 that is strictly greater than a given integer.

\[
\begin{align*}
10: & \quad 8A00 \quad R[A] <- M[00] \quad \text{load } a \text{ from } M[00] \\
11: & \quad 7101 \quad R[1] <- 1 \quad \text{power }= 1 \\
12: & \quad 221A \quad R[2] <- R[1] - R[A] \quad \text{while } (\text{power }<= a) \\
13: & \quad D216 \quad \text{if } (R[2] > 0) \quad \text{PC } <- 16 \\
14: & \quad 1111 \quad R[1] <- R[1] + R[1] \quad \text{power }= 2\times \text{power} \\
15: & \quad C012 \quad \text{goto 12} \\
16: & \quad 9101 \quad M[01] <- R[1] \quad \text{store power to } M[01] \\
17: & \quad 0000 \quad \text{halt}
\end{align*}
\]

(a) 0002

(b) 0008

(c) 0010

Remember that everything is in hex: \(10_{16} = 16_{10}\).

(d) 2000

You will get this part only by reasoning about what the TOY program does (or wasting an extraordinary amount of time tracing code).

(e) 0001

FACE is a negative integer (two’s complement representation). Thus, the loop is skipped and \(R[1]\) remains 0001.