There are eight questions on this exam, weighted as indicated at the bottom of this page. There is one question per lecture, numbered corresponding to the lectures, not in order of difficulty. If a question seems difficult to you, skip it and come back to it.

**Policies.** The exam is closed book, though you are allowed to use a single-page one-sided hand-written cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. You have 50 minutes to complete the test. **This exam is preprocessed by computer.** If you use pencil (and eraser), write darkly. Write all answers inside the designated rectangles. Do not write on corner marks.

**This page.** Print your name, NetID; fill in your precept on this page; write and sign the Honor Code pledge.

**Discussing this exam.** As you know, discussing the contents of this exam before solutions have been posted is a serious violation of the Honor Code.

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

---

**SOLUTION**

<table>
<thead>
<tr>
<th>Precept</th>
<th>Name</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>Dan Leyzberg</td>
<td>T/Th 12:30pm</td>
</tr>
<tr>
<td>P01A</td>
<td>Ted Brundage</td>
<td>T/Th 12:30pm</td>
</tr>
<tr>
<td>P01B</td>
<td>Doug Clark</td>
<td>T/Th 12:30pm</td>
</tr>
<tr>
<td>P02</td>
<td>Jérémie Lumbraso</td>
<td>T/Th 1:30pm</td>
</tr>
<tr>
<td>P02A</td>
<td>Doug Clark</td>
<td>T/Th 1:30pm</td>
</tr>
<tr>
<td>P02B</td>
<td>Andrea LaPaugh</td>
<td>T/Th 1:30pm</td>
</tr>
<tr>
<td>P03</td>
<td>Jérémie Lumbraso</td>
<td>T/Th 2:30pm</td>
</tr>
<tr>
<td>P04</td>
<td>Ming-Yee Tsang</td>
<td>T/Th 7:30pm</td>
</tr>
<tr>
<td>P05</td>
<td>Stephen Cook</td>
<td>W/F 1:30pm</td>
</tr>
<tr>
<td>P05A</td>
<td>Lawrence Lin</td>
<td>W/F 1:30pm</td>
</tr>
<tr>
<td>P06</td>
<td>Alan Kaplan</td>
<td>W/F 2:30pm</td>
</tr>
<tr>
<td>P11</td>
<td>Dan Leyzberg</td>
<td>T/Th 1:30pm</td>
</tr>
<tr>
<td>P12</td>
<td>Donna Gabai</td>
<td>W/F 1:30pm</td>
</tr>
<tr>
<td>P13</td>
<td>Alan Kaplan</td>
<td>T/Th 3:00pm</td>
</tr>
<tr>
<td>P14</td>
<td>Donna Gabai</td>
<td>W/F 3:00pm</td>
</tr>
</tbody>
</table>
Q0. Prologue (8 points). Consider the Boolean $\text{EQ}$, defined by the following truth table:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>a EQ b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

A. (3 points) In the box below, write the bit sequence that results by applying $\text{EQ}$ in a bitwise fashion to the two given bit sequences.

```
1 0 1 1 0 0 1 1 0 0 1 1 0 0 1
0 1 0 0 1 1 1 1 0 0 1 1 0 0 0
```

B. (Each question: 1pt if correct, -1pt if incorrect, 0pts if blank) Indicate whether each of the following statements are True or False, by filling the corresponding bubble.

- If $a \text{ EQ } b$ is 1 and $b \text{ EQ } c$ is 1, then $a \text{ EQ } c$ is 1.
  - True  False

- $\text{EQ}$ is equivalent to $\text{NOT XOR}$.
  - True  False

- The bitwise $\text{EQ}$ of a bitstring consisting of all 0s with a random bitstring will have about the same number of 0s as 1s.
  - True  False

- The bitwise $\text{EQ}$ of any bitstring with a random bitstring will have about the same number of 0s as 1s.
  - True  False

- The $\text{EQ}$ function could be substituted for $\text{XOR}$ in the encryption/decryption scheme described in lecture, without any ill effects.
  - True  False
Q1. Types and Casts (9 points). Give the value of each of the following Java expressions. If an expression will not compile or will cause an exception at runtime, put an X under value. If the value is a string, enclose it in double quotes. You must fill in every entry. Entries left blank will be marked incorrect.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / 0</td>
<td>X</td>
</tr>
<tr>
<td>&quot;800&quot; * 1</td>
<td>X</td>
</tr>
<tr>
<td>&quot;1&quot; + &quot; - &quot; + &quot;1&quot;</td>
<td>“1 - 1”</td>
</tr>
<tr>
<td>3.14159 + (int) Math.PI</td>
<td>6.14159</td>
</tr>
<tr>
<td>1 - 1 - 1 - 1</td>
<td>-2</td>
</tr>
<tr>
<td>3 / 2.0 + 2 * 5</td>
<td>11.5</td>
</tr>
<tr>
<td>(8 &lt;= 2)</td>
<td></td>
</tr>
<tr>
<td>Double.parseDouble(&quot;8.5*2&quot;)</td>
<td>X</td>
</tr>
<tr>
<td>&quot;1&quot; + 1 + 1 + &quot;1&quot;</td>
<td>“1111”</td>
</tr>
</tbody>
</table>
Q2. Loops and Arrays (9 points). Consider the following Java code fragment:

```java
int[] a = new int[N];
a[0] = 1;
for (int i = 1; i < N; i++)
    {
        int sum = 0;
        for (int j = 0; j < i; j++)
            sum = sum + a[j];
        a[i] = 1 + (2 * sum) / i;
    }
```

A. (6 points) Suppose that N is greater than 4. Fill in the following trace of the values of sum and a[i] just after each of the first four iteration of the outer for loop.

<table>
<thead>
<tr>
<th>i</th>
<th>sum</th>
<th>a[i]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>9</td>
</tr>
</tbody>
</table>

B. (3 points) In the box below, write one line of code that can be substituted for the four lines of code in the outer for loop above (lines 5 to 8) that would fill the array with the same values.

Valid solutions:

- \( a[i] = 2i + 1; \)
- \( a[i] = i + i + 1; \)
- \( a[i] = a[i-1] + 2; \)
Q3. More Arrays (9 points).

A. (3 points) In the box below, write one line of Java code that creates a 4-by-4 (two-dimensional) array named x of double values and initializes them each to the value 0.0.

Valid solutions:

```java
double[][] x = new double[4][4];
double[][] x = {{0.0, 0.0, 0.0, 0.0}, {0.0, 0.0, 0.0, 0.0},
                {0.0, 0.0, 0.0, 0.0}, {0.0, 0.0, 0.0, 0.0}};
```

B. (Each question: 1pt if correct, -1pt if incorrect, 0pts if blank) Indicate whether each of the following statements are True or False, by filling the corresponding bubble.

Once it has been created, you can change the size of a Java array.  
True  False

Once a one-dimensional array has been created and initialized, you can refer to the last element in the array with the code `a[a.length]`.  
True  False

Once a one-dimensional array has been created and initialized, you can refer to the first element in the array with the code `a[1]`.  
True  False

You can use an array as the return type of a function.  
True  False

If you pass an array as an argument to a function, the function can only read the array’s entries, not change them.  
True  False

If a and b are arrays of the same length, then the code `a = b` copies each element of b to the corresponding in a.  
True  False
Q4. Input and Output (9 points). Our task is to develop a program Q4.java that takes an integer N from the command line and then reads N point coordinates from standard input and draws the polygon formed by drawing lines connecting successive points, then a line connecting the Nth point to the first. For example, if the file Q4ex.txt contains the values listed at left below, then the command java Q4 3 < Q4ex.txt should draw a right triangle, and the command java Q4 4 < Q4ex.txt should draw a square, as depicted at right.

```
Q4ex.txt  java Q4 3 < Q4ex.txt  java Q4 4 < Q4ex.txt
0.1 0.2
0.8 0.2
0.8 0.9
0.1 0.9
```

The code below is a solution to this problem, with four snippets of code missing. Fill in the boxes with the code needed to complete this program.

YOU MAY WISH TO SAVE THIS QUESTION FOR LAST—IT MAY BE TIME-CONSUMING.

```java
public class Q4 {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        double xFirst, yFirst, xOld, yOld, xNew, yNew;
        xFirst = StdIn.readDouble();
        yFirst = StdIn.readDouble();
        xOld = xFirst;
        yOld = yFirst;
        for (int i = 1; i < N; i++) {
            xNew = StdIn.readDouble();
            yNew = StdIn.readDouble();
            StdDraw.line(xOld, yOld, xNew, yNew);
            xOld = xNew;
            yOld = yNew;
        }
        StdDraw.line(xNew, yNew, xFirst, yFirst);
    }
}
```
Q5. Functions (9 points). Consider the following Java program, which has no purpose other than to test your understanding of functions and scope:

```java
public class Q5 {
    public static int f(int x, int[] y, int[] z) {
        x = 1;
        y[0] = x + 1;
        z = new int[4];
        z[3] = 3;
        return z[3];
    }

    public static void main(String[] args) {
        int x = 0;
        int[] y = { 1, 2, 3, 4, 5 };
        int[] z = { 6, 7, 8, 9 };
        z[0] = f(x, y, z);
        System.out.println(x);
        System.out.println(y[0]);
        System.out.println(z[3]);
        System.out.println(z[0]);
    }
}
```

Fill in the requested bubbles.

First value printed

```
0 1 2 3 4 5 6 7 8 9 10
```

0 0 0 0 0 0 0 0 0 0 0

Second value printed

```
0 1 2 3 4 5 6 7 8 9 10
```

0 0 0 0 0 0 0 0 0 0 0

Third value printed

```
0 1 2 3 4 5 6 7 8 9 10
```

0 0 0 0 0 0 0 0 0 0 0

Fourth value printed

```
0 1 2 3 4 5 6 7 8 9 10
```

0 0 0 0 0 0 0 0 0 0 0
Q6. Recursion (9 points). Consider the following recursive static method:

```java
static int mystery(int a, int b)
{
    if (b == 0) return 0;
    if (a == 0) return mystery(b - 1, a);
    return b + mystery(b, a - 1);
}
```

For each call listed below, write the value returned in the box at the right. (The last response is worth 2 points; the others are 1 point each.)

<table>
<thead>
<tr>
<th>Call</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mystery(0, 0)</td>
<td>0</td>
</tr>
<tr>
<td>mystery(10, 0)</td>
<td>0</td>
</tr>
<tr>
<td>mystery(0, 10)</td>
<td>0</td>
</tr>
<tr>
<td>mystery(1, 10)</td>
<td>10</td>
</tr>
<tr>
<td>mystery(10, 1)</td>
<td>10</td>
</tr>
<tr>
<td>mystery(2, 10)</td>
<td>20</td>
</tr>
<tr>
<td>mystery(10, 3)</td>
<td>30</td>
</tr>
<tr>
<td>mystery(200, 300)</td>
<td>60000</td>
</tr>
</tbody>
</table>
Q7. Performance (8 points). Consider the following tables, which give experimental running times in seconds for four programs A, B, C, and D for various values of the input size N.

<table>
<thead>
<tr>
<th>N</th>
<th>running time</th>
<th>N</th>
<th>running time</th>
<th>N</th>
<th>running time</th>
<th>N</th>
<th>running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>21</td>
<td>100</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>2,000</td>
<td>80</td>
<td>1,000</td>
<td>25</td>
<td>10</td>
<td>3</td>
<td>200</td>
<td>25</td>
</tr>
<tr>
<td>4,000</td>
<td>325</td>
<td>10,000</td>
<td>260</td>
<td>20</td>
<td>10</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>8,000</td>
<td>1,275</td>
<td>100,000</td>
<td>2,600</td>
<td>40</td>
<td>100</td>
<td>800</td>
<td>1,599</td>
</tr>
</tbody>
</table>

To the right of each option, fill the bubble of the one-word hypothesis on the order of growth of the running time that best explains the given experimental evidence.

- linear
- quadratic
- cubic
- exponential

A: O  O  O  O
B: O  O  O  O
C: O  O  O  O
D: O  O  O  O