Assume that $x$ and $y$ are `int` variables. Consider the following code fragment:

```java
if (y > 3)
    x = x + 2;
if (y < 9)
    x = x + 3;
else
    x = x + 6;
System.out.print(x);
```

What is printed if $x$ is 0 and $y$ is 1 before the code fragment executes?

A. 2  
B. 3  
C. 5  
D. 6  
E. 8
Ifs, Switches, and Bools for 200

if (n == 5 || n == 7)
    System.out.println("yes");
else
    System.out.println("no");

Which is the best translation of the above code fragment?

A. switch (n) {
    case 5: System.out.println("yes");
    case 7: System.out.println("yes");
    default: System.out.println("no");
}

B. switch (n) {
    case 5:
    case 7: System.out.println("yes");
    break;
}
System.out.println("no");

C. switch (n) {
    case 5:
    case 7: System.out.println("yes");
    break;
    default: System.out.println("no");
}

D. switch (n) {
    case 5: System.out.println("yes");
    break;
    case 7: System.out.println("yes");
    break;
    default: System.out.println("no");
}
Ifs, Switches, and Bools for 400

Assume that `hour` and `min` have been declared to be `int` variables and that each has been initialized. Which of the following boolean expressions evaluates to true if and only if `hour:min` is a valid time between 4:12 and 5:37?

A. `(hour == 4 && min >= 12) || (hour == 5 && min <= 37)`

B. `(hour == 4 || hour == 5) && min >= 12 && min <= 37`

C. `(hour == 4 || min >= 12 || min <= 59) && (hour == 5 || min >= 0 || min <= 37)`

D. `(hour == 4 || hour == 5) && ((min >= 12 && min <= 59) || (min >= 0 && min <= 37))`

E. `(hour == 4 && min >= 12 && min <= 59) || (hour == 5 && min >= 0 && min <= 37)`
The above flow diagram is best described as which one of the following?

A. an if statement with a do-while loop nested inside

B. an if statement with a while loop nested inside

C. a do-while loop with an if statement nested inside

D. a while loop with a do-while loop nested inside

E. a while loop with an if statement nested inside
Loops for 200

Consider writing code that works as follows: *the user enters a positive number, and then that many strings are read.*

Which is the best kind of loop to use, and the correct reason for that choice?

A. a *while* loop because the code must read inputs until a condition is satisfied

B. a *do–while* loop so that both the positive number and the inputs can be read inside the loop

C. a *do–while* loop because at least one input must be read before looping

D. a *for* loop because the code must read inputs until a condition is satisfied

E. a *for* loop because the positive number tells how many times to loop
Loops for 400

Consider the following code fragment:

```java
int x = 0, y = 0;
while (x < 8) {
    y = 3;
    do {
        System.out.print('*');
        y--;
    } while (x <= y);
    x++;
}
```

How many stars are printed when this code executes?

A. 4
B. 8
C. 10
D. 11
E. 14
Expressions for 100

Assume that $s$ is a String whose length is at least 2. Which of the following boolean expressions correctly tests whether $s$ begins and ends with the same character (i.e., the first character of $s$ is the same as the last character of $s$)?

A. $s$.charAt(0) == $s$.charAt($s$.length())

B. $s$.charAt(0).equals($s$.charAt($s$.length()))

C. $s$.charAt(0) == $s$.charAt($s$.length()-1)

D. $s$.charAt(0).equals($s$.charAt($s$.length()-1))

E. $s$.charAt(1) == $s$.charAt($s$.length())
Expressions for 200

Assume that the following variable declarations have been made and that the variables have been initialized.

```java
int j, k;
double d1, d2;
boolean b;
```

Which of the following statements would not compile?

A. `j += k * 2;`

B. `d1 = (double)k/d2;`

C. `j = k * b;`

D. `b = !b;`

E. `b = (j == k);`
Assume that the following variable declarations and initializations have been made.

```java
int x = 2, y = 5;

double p = 3.0, q = 8.0;
```

The value of the following Java expression:

```
p * 2 + y / x + x / q
```

would be which one of the following?

A. 5.75
B. 6.25
C. 8.00
D. 8.25
E. 8.75
Arrays for 100

How can we complete the following code so that it returns true iff the int values in 1D array A are in sorted order?

```java
for (int k=0; k<A.length-1; k++) {
    if (condition) return value 1;
}
return value 2;
```

<table>
<thead>
<tr>
<th>condition</th>
<th>value 1</th>
<th>value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. A[k] &lt; A[k+1]</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>C. A[k] == A[k+1]</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>
Arrays for 200

```java
int[] B = new int[2*A.length];
int j = B.length-1;
for (int k=0; k<A.length; k++) {
    B[k] = A[k];
    B[j] = A[k];
    j--;
}
```

If A looks like this: [ 1 2 3 4 ], what does B look like after the code above finishes?

A. [ 1 2 3 4 4 3 2 1 ]
B. [ 1 2 3 4 1 2 3 4 ]
C. [ 4 3 2 1 1 2 3 4 ]
D. [ 4 3 2 1 4 3 2 1 ]
E. [ 1 4 2 3 1 4 2 3 ]
Arrays for 400

A is a 2D array. When does this code return true?

```java
for (int row=1; row<A.length; row++) {
    if (A[row][A[row].length-1] <=
        A[row-1][A[row-1].length-1]) {
        return false;
    }
}
return true;
```

A. iff the last value in every row of $A$ is larger than all other values in that row

B. iff the last value in every row of $A$ is larger than the last value in the previous row

C. iff the values in the last row of $A$ are larger than the values in all other rows

D. iff the values in the last column of $A$ are larger than the values in all other columns

E. iff the values in the last row of $A$ are larger than the values in the last column of $A$
Static Methods for 100

public static int getInt(Scanner sc) {
    System.out.println("Enter an int: ");
    return sc.nextInt();
}

Assume that variable in is a properly initialized Scanner. Which of the following statements would compile?

A. System.out.println(getInt(in));

B. Scanner sc = getInt(in);

C. int num = getInt(Scanner in);

D. System.out.println(getInt(5));

E. int num = getInt(in, 5);
What happens when this code is compiled and run?

A. The code does not compile because mystery changes its parameter.

B. The code compiles, but there is a run-time error because mystery changes its parameter.

C. 20 is printed.

D. 21 is printed.

E. 22 is printed.
public static boolean larger(int x, int y) {
    if (x > y) return true;
    if (y > x) return false;
}

Which of the following statements about method larger is correct?

A. The code will not compile because it does not always return a value.
B. The code will not compile because it has int parameters but returns a boolean.
C. The code will compile and always returns the correct value.
D. The code will compile but always returns the wrong value.
E. The code will compile but sometimes returns the wrong value.
char[][] B = new char[A.length][A.length];
for (int row=0; row<A.length; row++) {
    int k = B.length-(row+1);
    for (int col=0; col<A.length; col++) {
        switch (A[row][col]) {
            case '-': B[k][col] = 'o';
                break;
            case '*':
                break;
            case '+': B[k][col] = 'x';
                break;
            default: B[k][col] = ' ';
        }
    }
}

If A looks like this

```
+ +
+ +
- - - -
- -
```

What does B look like after the code executes?