CS10-2DArray-Quiz - Version 1

Question 1. Consider the following code segment.

```
int[][] grid = {{4, 9, 2},
                {7, 1, 8},
                {3, 5, 6},
                {2, 4, 0}};
int result = 0;
for (int c = 0; c < grid[0].length; c++)
{
   for (int r = 0; r < grid.length; r++)
   {
      if (grid[r][c] > result)
      {
         result = grid[r][c];
      }
   }
}
```

Which of the following best describes the value stored in `result` as a result of executing the code segment?

A) The column index of the largest value in `grid`
B) The largest value in the last row of `grid`
C) The sum of the largest value in each column of `grid`
D) The largest value in the first column of `grid`
E) The largest value among all elements of `grid`

Question 2. Consider the following 2D array declaration.

```
int[][] temps = {{5, 7, -2},
                 {-9, 3, -6},
                 {-1, 8, -5},
                 {-3, 9, -7}};
```

Which of the following code segments prints the largest value in each column of `temps`, from the leftmost column to the rightmost column, with each value followed by a space?

A)
```
for (int r = 0; r < temps.length; r++)
{
   int max = temps[r][0];
   for (int c = 1; c < temps[r].length; c++)
   {
      if (temps[r][c] > max)
      {
         max = temps[r][c];
      }
   }
   System.out.print(max + " ");
}
```

B)
```
for (int c = 0; c < temps[0].length; c++)
{
   int max = temps[0][c];
   for (int r = 1; r < temps.length; r++)
   {
      if (temps[r][c] > max)
      {
         max = temps[r][c];
      }
   }
   System.out.print(max + " ");
}
```

C)
```
for (int c = 0; c < temps[0].length; c++)
{
   int max = temps[0][0];
   for (int r = 0; r < temps.length; r++)
   {
      if (temps[r][c] > max)
      {
         max = temps[r][c];
      }
   }
   System.out.print(max + " ");
}
```

D)
```
for (int c = 0; c < temps[0].length; c++)
{
   int max = 0;
   for (int r = 0; r < temps.length; r++)
   {
      if (temps[r][c] > max)
      {
         max = temps[r][c];
      }
   }
   System.out.print(max + " ");
}
```

Question 3. Consider the following code segment.

```
int[][] scores = {{4, 7, 2}, {9, 1, 5}};
int total = 0;

for (int c = 0; c < scores.length; c++)
{
   total += scores[c][1];
}

System.out.println(total);
```

The code segment is intended to print the sum of all values in row 1 of `scores`. Which of the following modifications will cause the code segment to print the correct sum?

A) Change `total += scores[c][1];` to `total += scores[1][c];` and change the loop condition to `c <= scores[1].length`.
B) Change `total += scores[c][1];` to `total += scores[c][c];` and leave the loop condition as `c < scores.length`.
C) Change the loop condition to `c < scores[1].length` and leave `total += scores[c][1];` unchanged.
D) Change `total += scores[c][1];` to `total += scores[1][c];` and change the loop condition to `c < scores[1].length`.
E) Change `total += scores[c][1];` to `total += scores[1][c];` and leave the loop condition as `c < scores.length`.

Question 4. Consider the following code segment.

```
int[][] table = new int[3][5];
for (int r = 0; r < table.length; r++)
{
   for (int c = 0; c < table[r].length; c++)
   {
      table[r][c] = (r + 1) * c - r;
   }
}
System.out.println(table[1][4] + table[2][2]);
```

What is printed as a result of executing the code segment?

A) 6
B) 8
C) 4
D) 14
E) 11

Question 5. Consider the following code segment.

```
int[][] grid = {{3, 8, 1, 6},
                {7, 2, 9, 4},
                {5, 0, 6, 2}};
int count = 0;
for (int r = 0; r < grid.length; r++)
{
   for (/* missing inner loop header */)
   {
      if (grid[r][c] > 5)
      {
         count++;
      }
   }
}
```

The code segment is intended to store in `count` the number of elements in `grid` whose value is greater than 5. Which of the following should replace `/* missing inner loop header */` so that the code segment works as intended?

A) `int c = 0; c < grid[r].length; c++`
B) `int c = 1; c < grid[r].length; c++`
C) `int c = 2; c < grid[r].length; c++`
D) `int c = 0; c <= grid[r].length; c++`
E) `int c = 0; c < grid.length; c++`

Question 6. Consider the following 2D array declaration.

```
int[][] data = {{4, -3, 7, 1},
                {2, 9, -8, 6},
                {-1, 5, 6, -4}};
```

Which of the following code segments prints the row index and column index of each negative value in `data`, one pair per line? Each line should contain the row index, then a single space, then the column index.

A)
```
for (int r = 0; r < data.length; r++)
{
   for (int c = 0; c < data[r].length; c++)
   {
      if (data[r][c] < 0)
      {
         System.out.println(r + " " + c);
      }
   }
}
```

B)
```
for (int r = 0; r < data.length; r++)
{
   for (int c = 0; c < data[r].length; c++)
   {
      if (data[r][c] < 0)
      {
         System.out.println(c + " " + r);
      }
   }
}
```

C)
```
for (int r = 0; r < data.length; r++)
{
   for (int c = 0; c < data[r].length; c++)
   {
      System.out.println(r + " " + c);
   }
}
```

D)
```
for (int r = 0; r < data.length; r++)
{
   for (int c = 0; c < data.length; c++)
   {
      if (data[r][c] < 0)
      {
         System.out.println(r + " " + c);
      }
   }
}
```

Question 7. Consider the following code

```
public static int mystery(int[][] arr)
{
   int count = 0;
   for (int r = 0; r < arr.length; r++)
   {
      for (int c = 0; c < arr[r].length; c++)
      {
         if (arr[r][c] % 2 != 0)
         {
            count++;
         }
      }
   }
   return count;
}
```

Consider the following code segment.

```
int[][] data = {{4, -3, 7, 2},
                {10, 5, -8, 6},
                {-1, 0, 9, 12}};

System.out.println(mystery(data));
```

What is printed as a result of executing the code segment?

A) 2
B) 3
C) 12
D) 7
E) 5

Question 8. Consider the following 2D array declaration.

```
int[][] nums = {{2, 5, 1},
                {4, 0, 3},
                {6, 7, 8},
                {1, 9, 2}};
```

Which of the following code segments prints the sum of each column of nums, from the leftmost column to the rightmost column, with each sum printed on its own line?

A)
```
for (int c = 0; c < nums[0].length; c++)
{
   int colSum = 0;
   for (int r = 0; r < nums.length; r++)
   {
      colSum += nums[c][r];
   }
   System.out.println(colSum);
}
```

B)
```
int colSum = 0;
for (int r = 0; r < nums.length; r++)
{
   for (int c = 0; c < nums[0].length; c++)
   {
      colSum += nums[r][c];
   }
   System.out.println(colSum);
}
```

C)
```
for (int c = 0; c < nums[0].length; c++)
{
   int colSum = 0;
   for (int r = 0; r < nums.length; r++)
   {
      colSum += nums[r][c];
   }
   System.out.println(colSum);
}
```

D)
```
for (int r = 0; r < nums.length; r++)
{
   int rowSum = 0;
   for (int c = 0; c < nums[0].length; c++)
   {
      rowSum += nums[r][c];
   }
   System.out.println(rowSum);
}
```