chap 8: 2D Array

1. Consider the following code segment.

```
int[][] anArray = new int[10][8];

for (int j = 0; j < 8; j++)
{
     for (int k = 0; k < 10; k++)
     {
         anArray[j][k] = 5;
     }
}
```

The code segment causes an `ArrayIndexOutOfBoundsException` to be thrown. How many elements in `anArray` will be set to `5` before the exception is thrown?

(A) 0
(B) 8
(C) 9
(D) 64
(E) 80

chap 8: 2D Array

2. Assume mat is defined as follows.

```
int dim = 4;

int[][] mat = new int[dim][dim];
```

Consider the following code segment.

```
int sum = 0;

for (int row = 0; row < dim; row++)

{

  sum = sum + mat[row][dim - 1];

}
```

Assume that mat contains the following values before the code segment is executed. Note that mat[0][3] is 2.

	0	**1**	**2**	**3**
0	1	1	2	2
1	1	2	2	4
2	1	3	2	6
3	1	4	2	8

What value will sum contain after the code segment is executed?

(A) 6

(B) 8

(C) 13

(D) 15

(E) 20

chap 8: 2D Array

3. Assume that `mat` has been declared as a 4×4 array of integers and has been initialized to contain all `1`s. Consider the following code segment.

```
int n = mat.length;
for (int j = 1; j < n; j++)
{
     for (int k = 1; k < n; k++)
     {
         mat[j][k] = mat[j - 1][k] + mat[j][k - 1];
     }
}
```

What is the value of `mat[2][2]` after the code segment has completed execution?

(A) `2`

(B) `3`

(C) `4`

(D) `6`

(E) `10`

chap 8: 2D Array

4. Consider the following code segment.

```
int[][] mat = new int[3][4];
for (int row = 0; row < mat.length; row++)
{
   for (int col = 0; col < mat[0].length; col++)
   {
     if (row < col)
     {
       mat[row][col] = 1;
     }
     else if (row == col)
     {
       mat[row][col] = 2;
     }
     else
     {
       mat[row][col] = 3;
     }
   }
}
```

What are the contents of mat after the code segment has been executed?

chap 8: 2D Array

(A)
```
{{2, 1, 1},
 {3, 2, 1},
 {3, 3, 2},
 {3, 3, 3}}
```

(B)
```
{{2, 3, 3},
 {1, 2, 3},
 {1, 1, 2},
 {1, 1, 1}}
```

(C)
```
{{2, 3, 3, 3},
 {1, 2, 3, 3},
 {1, 1, 2, 3}}
```

(D)
```
{{2, 1, 1, 1},
 {3, 2, 1, 1},
 {3, 3, 2, 1}}
```

(E)
```
{{1, 1, 1, 1},
 {2, 2, 2, 2},
 {3, 3, 3, 3}}
```

5. Consider the following code segment.

```
String[][] letters = {{"A", "B", "C", "D"},
                      {"E", "F", "G", "H"},
                      {"I", "J", "K", "L"}};
for (int col = 1; col <  letters[0].length; col++)
{
     for (int row = 1; row <  letters.length; row++)
     {
         System.out.print(letters[row][col] + " ");
     }
     System.out.println();
}
```

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

chap 8: 2D Array

(A)
```
A E I
F J
K
```

(B)
```
B F J
C G K
D H L
```

(C)
```
E I
F J
G K
H L
```

(D)
```
F G H
J K L
```

(E)
```
F J
G K
H L
```

6. A two-dimensional array `arr` is to be created with the following contents.

```
boolean[][] arr = {{false, true, false},
                   {false, false, true}};
```

Which of the following code segments can be used to correctly create and initialize `arr` ?

(A)
```
boolean arr[][] = new boolean[2][3];
arr[0][1] = true;
arr[1][2] = true;
```

(B)
```
boolean arr[][] = new boolean[2][3];
arr[1][2] = true;
arr[2][3] = true;
```

(C)
```
boolean arr[][] = new boolean[3][2];
arr[0][1] = true;
arr[1][2] = true;
```

(D)
```
boolean arr[][] = new boolean[3][2];
arr[1][0] = true;
arr[2][1] = true;
```

(E)
```
boolean arr[][] = new boolean[3][2];
arr[2][1] = true;
arr[3][2] = true;
```

chap 8: 2D Array

7. Consider the following method.

```
/** Precondition: values has at least one row */
public static int calculate(int[][] values)
{
  int found = values[0][0];
  int result = 0;
  for (int[] row : values)
  {
    for (int y = 0; y < row.length; y++)
    {
      if (row[y] > found)
      {
        found = row[y];
        result = y;
      }
    }
  }
  return result;
}
```

Which of the following best describes what is returned by the calculate method?

(A) The largest value in the two-dimensional array

(B) The smallest value in the two-dimensional array

(C) The row index of an element with the largest value in the two-dimensional array

(D) The row index of an element with the smallest value in the two-dimensional array

(E) The column index of an element with the largest value in the two-dimensional array

chap 8: 2D Array

8. Consider the following method, which is intended to return the number of columns in the two-dimensional array `arr` for which the sum of the elements in the column is greater than the parameter `val`.

```
public int countCols(int[][] arr, int val)
{
     int count = 0;

     for (int col = 0; col < arr[0].length; col++) // Line 5
     {
         int sum = 0;
         for (int[] row : col) // Line 8
         {
            sum += row[col]; // Line 10
         }
         if (sum > val)
         {
            count++;
         }
     }
     return count;
}
```

The `countCols` method does not work as intended. Which of the following changes should be made so the method works as intended?

(A) Line 5 should be changed to `for (int col = 0; col < arr.length; col++)`

(B) Line 8 should be changed to `for (int row : col)`

(C) Line 8 should be changed to `for (int[] row : arr)`

(D) Line 10 should be changed to `sum += arr[col];`

(E) Line 10 should be changed to `sum += arr[row][col];`

9. Consider the following code segment, which is intended to declare and initialize the two-dimensional (2D) `String` array `things`.

```
/* missing code */ = {{"spices", "garlic", "onion", "pepper"},
                      {"clothing", "hat", "scarf", "gloves"},
                      {"plants", "tree", "bush", "flower"},
                      {"vehicles", "car", "boat", "airplane"}};
```

Which of the following could replace `/* missing code */` so that `things` is properly declared?

(A) `new String[][] things`

(B) `new(String[][]) things`

(C) `String[] String[] things`

(D) `String[][] things`

(E) `[][]String things`

chap 8: 2D Array

10. Consider the following method, `count`, which is intended to traverse all the elements in the two-dimensional (2D) `String` array `things` and return the total number of elements that contain at least one `"a"`.

```
public static int count(String[][] things)
{
     int count = 0;
     for (int r = 0; r < things.length; r++)
     {
         for (int c = 0; c < things[r].length - 1; c++)
         {
            if (things[r][c].indexOf("a") >= 0)
            {
           count++;
            }
         }
     }
     return count;
}
```

For example, if `things` contains `{{"salad", "soup"}, {"water", "coffee"}}`, then `count(things)` should return `2`.

The method does not always work as intended. For which of the following two-dimensional array input values does `count` NOT work as intended?

(A) `{{"lemon"}, {"lime"}}`

(B) `{{"tall", "short"}, {"up", "down"}}`

(C) `{{"rabbit", "bird"}, {"cat", "dog"}, {"gecko", "turtle"}}`

(D) `{{"scarf", "gloves", "hat"}, {"shoes", "shirt", "pants"}}`

(E) `{{"math", "english", "physics"}, {"golf", "baseball", "soccer"}}`

chap 8: 2D Array

11. Consider the following method, which is intended to return `true` if `0` is found in its two-dimensional array parameter `arr` and `false` otherwise. The method does not work as intended.

```
public boolean findZero(int[][] arr)
{
     for (int row = 0; row <= arr.length; row++)
     {
         for (int col = 0; col < arr[0].length; col++)
         {
            if (arr[row][col] == 0)
            {
           return true;
            }
         }
     }
     return false;
}
```

Which of the following values of `arr` could be used to show that the method does not work as intended?

(A) `{{30, 20}, {10, 0}}`

(B) `{{4, 3}, {2, 1}, {0, -1}}`

(C) `{{0, 1, 2}, {3, 4, 5}, {6, 7, 8}}`

(D) `{{5, 10, 15, 20}, {25, 30, 35, 40}}`

(E) `{{10, 20, 0, 30, 40}, {60, 0, 70, 80, 90}}`

12. Consider the following code segment, where `num` is a properly declared and initialized integer variable. The code segment is intended to traverse a two-dimensional (2D) array `arr` looking for a value equal to `num` and then print the value. The code segment does not work as intended.

```
int[][] arr = {{7, 3, 6, 4},
          {9, 2, 0, 5},
          {1, 4, 3, 8}};
for (int j = 0; j < arr.length - 1; j++)
{
     for (int k = 0; k < arr[0].length; k++)
     {
         if (arr[j][k] == num)
         {
            System.out.println(arr[j][k]);
         }
     }
}
```

For which of the following values of `num` does the code segment not work as intended?

chap 8: 2D Array

(A) `num = 5`

(B) `num = 6`

(C) `num = 7`

(D) `num = 8`

(E) `num = 9`

13. Consider the following code segment.

```
int[][] arr = {{6, 2, 5, 7},
               {7, 6, 1, 2}};
for (int j = 0; j < arr.length; j++)
{
     for (int k = 0; k < arr[0].length; k++)
     {
          if (arr[j][k] > j + k)
          {
             System.out.println("!");
          }
     }
}
```

How many times will `"!"` be printed when the code segment is executed?

(A) 0 times

(B) 2 times

(C) 4 times

(D) 6 times

(E) 8 times

14. Consider the following two-dimensional array definition.

```
int[][] data = new int[5][10];
```

Consider the following code segment, where all elements in `data` have been initialized.

```
for (int j = 0; j < data.length; j++)
{
     for (int k = 0; k < data[0].length; k++)
     {
          if (j == k)
          {
             System.out.println(data[j][k]);
          }
     }
}
```

How many times is the `println` method called when the code segment is executed?

chap 8: 2D Array

(A) 4
(B) 5
(C) 9
(D) 10
(E) 15

15. Consider the following code segment.

```
int[][] array2D = {{1, 2, 3, 4},
                   {5, 6, 7, 8},
                   {9, 10, 11, 12},
                   {13, 14, 15, 16}};
for (int[] i : array2D)
{
     for (int x : i)
     {
         System.out.print(x + " ");
     }
     System.out.println(" ");
}
```

How many times will the statement `System.out.print(x + " ")` be executed?

(A) 3 times
(B) 4 times
(C) 6 times
(D) 12 times
(E) 16 times

chap 8: 2D Array

16. Consider the following data field and method.

```
private int[][] mat;

public void mystery ()
{
  for (int row = 1; row < mat.length; row++)
  {
    for (int col = 0; col < mat[0].length; col++)
    {
      if (row != col)
        mat[row][col] = mat[row - 1][col];
    }
  }
}
```

Assume that mat contains the following values. Note that mat[0][4] is 2.

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

What values does mat contain after a call to mystery?

chap 8: 2D Array

(A)
```
4 1 3 4 2
4 8 3 4 2
4 8 6 4 2
4 8 6 2 2
4 8 6 2 3
```

(B)
```
4 1 3 4 2
4 1 3 4 2
4 1 3 4 2
4 1 3 4 2
4 1 3 4 2
```

(C)
```
4 1 3 4 2
4 1 3 4 2
1 8 7 5 3
7 4 6 9 2
3 8 1 2 4
```

(D)
```
4 4 4 4 4
1 1 1 1 1
7 7 7 7 7
3 3 3 3 3
5 5 5 5 5
```

(E)
```
4 8 6 2 3
4 8 6 2 3
4 8 6 2 3
4 8 6 2 3
4 8 6 2 3
```

chap 8: 2D Array

17. Consider the following method.

```
public static int[] operation(int[][] matrix, int r, int c)
{
  int[] result = new int[matrix.length];

  for (int j = 0 ; j < matrix.length ; j++)
  {
    result[j] = matrix[r][j] * matrix[j][c];
  }
  return result;
}
```

The following code segment appears in another method in the same class.

```
int[][] mat = {{3, 2, 1, 4},
               {1, 2, 3, 4},
               {2, 2, 1, 2},
               {1, 1, 1, 1}};

int[] arr = operation(mat, 1, 2);
```

Which of the following represents the contents of arr as a result of executing the code segment?

(A) {6, 4, 2, 4}

(B) {1, 6, 3, 4}

(C) {4, 3, 6, 1}

(D) {4, 4, 2, 2}

(E) {2, 2, 4, 4}

18. Assume that a two-dimensional (2D) array `arr` of `String` objects with 3 rows and 4 columns has been properly declared and initialized.

Which of the following can be used to print the elements in the four corner elements of `arr` ?

(A) `System.out.print(arr[0, 0] + arr[0, 3] + arr[2, 0] + arr[2, 3]);`

(B) `System.out.print(arr[1, 1] + arr[1, 4] + arr[3, 1] + arr[3, 4]);`

(C) `System.out.print(arr[0][0] + arr[0][2] + arr[3][0] + arr[3][2]);`

(D) `System.out.print(arr[0][0] + arr[0][3] + arr[2][0] + arr[2][3]);`

(E) `System.out.print(arr[1][1] + arr[1][4] + arr[3][1] + arr[3][4]);`

chap 8: 2D Array

19. Consider the following code segment, where `letters` is a two-dimensional (2D) array that contains possible letters. The code segment is intended to print `"DIG"`.

```
String[][] letters = {{"A", "B", "C"},
                      {"D", "E", "F"},
                      {"G", "H", "I"}};
System.out.println( /* missing code */ );
```

Which of the following could replace `/* missing code */` so that the code segment works as intended?

(A) `letters[2][1] + letters[3][3] + letters[3][1]`

(B) `letters[2][0] + letters[2][2] + letters[1][0]`

(C) `letters[1][2] + letters[3][3] + letters[1][3]`

(D) `letters[1][0] + letters[2][2] + letters[2][0]`

(E) `letters[0][1] + letters[2][2] + letters[0][2]`

20. Consider the following code segment.

```
int[][] points = {{11, 12, 13, 14, 15},
                  {21, 22, 23, 24, 25},
                  {31, 32, 33, 34, 35},
                  {41, 42, 43, 44, 45}};
for (int row = 0; row < points.length; row++)
{
     for (int col = points[0].length - 1; col >= row; col--)
     {
         System.out.print(points[row][col] + " ");
     }
     System.out.println();
}
```

What is printed when this code segment is executed?

chap 8: 2D Array

(A)
```
15 14
25 24 23
35 34 33 32
45 44 43 42 41
```

(B)
```
15 14 13 12
25 24 23
35 34
45
```

(C)
```
11 12 13 14 15
21 22 23 24
31 32 33
41 42
```

(D)
```
15 14 13 12 11
25 24 23 22
35 34 33
45 44
```

(E)
```
15 14 13 12 11
25 24 23 22 21
35 34 33 32 31
45 44 43 42 41
```

21. Consider the following method, which is intended to print the values in its two-dimensional integer array parameter in row-major order.

```
public static void rowMajor(int[][] arr)
{
     /* missing code */
}
```

As an example, consider the following code segment.

```
int[][] theArray = {{1, 2}, {3, 4}, {5, 6}, {7, 8}};
rowMajor(theArray);
```

When executed, the code segment should produce the following output.

```
1 2 3 4 5 6 7 8
```

Which of the following code segments can replace `/* missing code */` so that the `rowMajor` method works as intended?

chap 8: 2D Array

(A)
```
for (int j : arr)
{
   for (int k : j)
   {
      System.out.print(j + " ");
   }
}
```

(B)
```
for (int j : arr)
{
   for (int k : j)
   {
      System.out.print(k + " ");
   }
}
```

(C)
```
for (int[] j : arr)
{
   for (int k : j)
   {
      System.out.print(j + " ");
   }
}
```

(D)
```
for (int[] j : arr)
{
   for (int k : j)
   {
      System.out.print(k + " ");
   }
}
```

(E)
```
for (int[] j : arr)
{
   for (int k : j)
   {
      System.out.print(arr[k] + " ");
   }
}
```

22. Consider the following code segment, where `nums` is a two-dimensional (2D) array of integers. The code segment is intended to print `"test1234"`.

```
System.out.print("test" + nums[0][0] + nums[1][0] + nums[1][1] +
nums[0][1]);
```

Which of the following code segments properly declares and initializes `nums` so that the code segment works as intended?

chap 8: 2D Array

(A) `int[][] nums = {{1, 2}, {3, 4}};`
(B) `int[][] nums = {{1, 2}, {4, 3}};`
(C) `int[][] nums = {{1, 3}, {2, 4}};`
(D) `int[][] nums = {{1, 4}, {2, 3}};`
(E) `int[][] nums = {{1, 4}, {3, 2}};`

23. Consider the following definition.

```
int[][] numbers = {{1, 2, 3},
                   {4, 5, 6}};
```

Which of the following code segments produces the output 123456 ?

chap 8: 2D Array

(A)
```
for (int[] row : numbers)
{
  for (int n : row)
  {
    System.out.print(n);
  }
}
```

(B)
```
for (int[] row : numbers)
{
  for (int n : row)
  {
    System.out.print(row[n]);
  }
}
```

(C)
```
for (int rc = 0; rc < numbers.length; rc++)
{
  System.out.print(numbers[rc]);
}
```

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

(E)
```
for (int c = 0; c < numbers[0].length; c++)
{
  for (int r = 0; r < numbers.length; r++)
  {
    System.out.print(numbers[r][c]);
  }
}
```

chap 8: 2D Array

24. Consider the following code segment, where `num` is an integer variable.

```
int[][] arr = {{11, 13, 14 ,15},
               {12, 18, 17, 26},
               {13, 21, 26, 29},
               {14, 17, 22, 28}};
for (int j = 0; j < arr.length; j++)
{
     for (int k = 0; k < arr[0].length; k++)
     {
         if (arr[j][k] == num)
         {
            System.out.print(j + k + arr[j][k] + " ");
         }
     }
}
```

What is printed when `num` has the value `14` ?

(A) `14 14`

(B) `16 17`

(C) `17 16`

(D) `18 19`

(E) `19 18`

25. Consider the following code segment.

```
String[][] arr = {{"Hello,", "Hi,", "Hey,"},
                  {"it's", "it is", "it really is"},
                  {"nice", "great", "a pleasure"},
                  {"to", "to get to", "to finally"},
                  {"meet", "see", "catch up with"},
                  {"you", "you again", "you all"}};
for (int j = 0; j < arr.length; j++)
{
     for (int k = 0; k < arr[0].length; k++)
     {
         if (k == 1)
         {
            System.out.print(arr[j][k] + " ");
         }
     }
}
```

What, if anything, is printed when the code segment is executed?

chap 8: 2D Array

(A) Nothing is printed due to an `ArrayIndexOutOfBoundsException`.

(B) `Hello, it's nice to meet you`

(C) `Hey, it really is a pleasure to finally catch up with you all`

(D) `Hi, it is great to get to see you again`

(E) `it's it is it really is`

26. Consider the following code segment.

```
int[][] values = {{1, 2, 3}, {4, 5, 6}};
int x = 0;
for (int j = 0; j < values.length; j++)
{
     for (int k = 0; k < values[0].length; k++)
     {
          if (k == 0)
          {
             values[j][k] *= 2;
          }
          x += values[j][k];
     }
}
```

What is the value of `x` after the code segment is executed?

(A) 7

(B) 17

(C) 21

(D) 26

(E) 27

27. Consider the following code segment.

```
int[][] arr = {{1, 2, 3, 4},
               {5, 6, 7, 8},
               {9, 10, 11, 12}};
int sum = 0;
for (int[] x : arr)
{
     for (int y = 0; y < x.length - 1; y++)
     {
          sum += x[y];
     }
}
```

What is the value of `sum` as a result of executing the code segment?

chap 8: 2D Array

(A) 36
(B) 54
(C) 63
(D) 68
(E) 78

28. Consider the following code segment.

```
int[] oldArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};
int[][] newArray = new int[3][3];

int row = 0;
int col = 0;
for (int value : oldArray)
{
  newArray[row][col] = value;
  row++;
  if ((row % 3) == 0)
  {
    col++;
    row = 0;
  }
}

System.out.println(newArray[0][2]);
```

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

(A) 3
(B) 4
(C) 5
(D) 7
(E) 8

chap 8: 2D Array

29. Consider the following code segment.

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

int[][] newArray = new int[3][3];

int row = 0; int col = 0;

for (int index = 0; index < oldArray.length; index++)

{

newArray[row][col] = oldArray[index]; row++;

if ((row % 3) == 0)

{

col++;

row = 0;

}

}

System.out.println(newArray[0][2]);
```

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

(A) 3
(B) 4
(C) 5
(D) 7
(E) 8

chap 8: 2D Array

30. Consider the following code segment.

```
int[][] arr = {{1, 2, 3},
               {4, 5, 6},
               {7, 8, 9},
               {3, 2, 1}};
for (int j = 0; j < arr.length; j++)
{
     for (int k = j; k < arr[0].length; k++)
     {
         System.out.print(arr[j][k] + " ");
     }
     System.out.println();
}
```

What output is printed when the code segment is executed?

(A)
```
2 3
6
```

(B)
```
1 2 3
4 5
7
```

(C)
```
1 2 3
5 6
9
```

(D)
```
1 4 7
5 8
9
```

(E)
```
1 2 3
5 6
9
1
```

chap 8: 2D Array

31. Consider the following Util class, which contains two methods. The completed sum1D method returns the sum of all the elements of the 1-dimensional array a. The incomplete sum2D method is intended to return the sum of all the elements of the 2-dimensional array m.

```
public class Util
{
  /** Returns the sum of the elements of the 1-dimensional array a */
  public static int sum1D(int[] a)
  {  /* implementation not shown */  }

  /** Returns the sum of the elements of the 2-dimensional array m */
  public static int sum2D(int[][] m)
  {
    int sum = 0;

    /* missing code */

    return sum;
  }
}
```

Assume that sum1D works correctly. Which of the following can replace / * *missing code* * / so that the sum2D method works correctly?

I.
```
for (int k = 0; k < m.length; k++)
{
  sum += sum1D(m[k]);
}
```

II.
```
for (int[] row : m)
{
  sum += sum1D(row);
}
```

III.
```
for (int[] row : m)
{
  for (int v : row)
  {
    sum += v;
  }
}
```

chap 8: 2D Array

(A) I only
(B) II only
(C) I and II only
(D) II and III only
(E) I, II, and III

32. Consider the following method.

```
public boolean checkIndexes(double[][] data, int row, int col)
{
     int numRows = data.length;
     if (row < numRows)
     {
         int numCols = data[0].length;
         return col < numCols;
     }
     else
     {
         return false;
     }
}
```

Consider the following variable declaration and initialization, which appears in a method in the same class as `checkIndexes`.

```
double[][] table = new double[5][6];
```

Which of the following method calls returns a value of `true` ?

(A) `checkIndexes(table, 4, 5)`
(B) `checkIndexes(table, 4, 6)`
(C) `checkIndexes(table, 5, 4)`
(D) `checkIndexes(table, 5, 6)`
(E) `checkIndexes(table, 6, 5)`

chap 8: 2D Array

33. Consider the following code segment.

```
String[][] board = new String[5][5];

for (int row = 0; row < 5; row++)
{
  for (int col = 0; col < 5; col++)
  {
    board[row][col] = "O";
  }
}

for (int val = 0; val < 5; val++)
{
  if (val % 2 == 1)
  {
    int row = val;
    int col = 0;
    while (col < 5 && row >= 0)
    {
      board[row][col] = "X";
      col++;
      row--;
    }
  }
}
```

Which of the following represents board after this code segment is executed?

chap 8: 2D Array

(A)

	0	1	2	3	4
0	X	O	X	O	X
1	O	X	O	X	O
2	X	O	X	O	X
3	O	X	O	X	O
4	X	O	X	O	X

(B)

	0	1	2	3	4
0	O	X	O	X	O
1	X	O	X	O	X
2	O	X	O	X	O
3	X	O	X	O	X
4	O	X	O	X	O

(C)

	0	1	2	3	4
0	X	O	O	O	X
1	O	X	O	X	O
2	O	O	X	O	O
3	O	X	O	X	O
4	X	O	O	O	X

(D)

	0	1	2	3	4
0	O	X	O	O	O
1	O	O	X	O	O
2	X	O	O	X	O
3	O	X	O	O	X
4	O	O	X	O	O

(E)

	0	1	2	3	4
0	O	X	O	X	O
1	X	O	X	O	O
2	O	X	O	O	O
3	X	O	O	O	O
4	O	O	O	O	O