Name: Dan Cassidy

Class: CSCI-C 490, Mobile Application Development

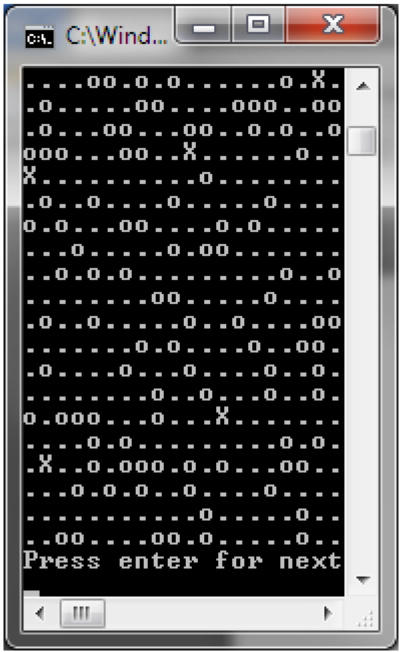
Assignment: Homework 3 Part 1

Date: 2015-07-10

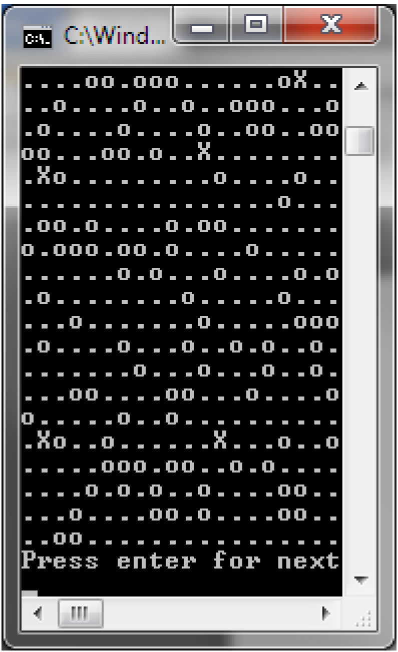
```java
/**
* Now define Anteater Class
*/
class Anteater extends Organism
{
    public static final int ANTEATER_BREED  = 8;
    public static final int ANTEATER_STARVE = 3;

    // Number of ticks since eating.
    private int starveTicks = 0;

    /**
     * Default constructor.
     */
    public Anteater()
    {
        super();
    }

    /**
     * 3-parameter constructor.
     *
     * @param world The World object this Anteater lives in.
     * @param x The x coordinate of the spot in <b>world</b> this Anteater inhabits.
     * @param y The y coordinate of the spot in <b>world</b> this Anteater inhabits.
     */
    public Anteater(World world, int x, int y)
    {
        super(world, x, y);
    }

    // Basic code reused from Ant.breed() method with changes to create an Anteater object instead
    // of an Ant object.

    /**
     * Adjusts the breed counter for the Anteater object and creates a new one when the appropriate
     * condition is met.
     */
    public void breed()
    {
        breedTicks++;
        if (breedTicks == ANTEATER_BREED)
        {
            breedTicks = 0;
            // Try to create a new Anteater object. Because world reference is passed in and
            // Anteater object adds itself to that world, Anteater reference doesn't need to be
            // explicitly saved.
            if ((y > 0) && (world.getAt(x, y - 1) == null))
            {
                new Anteater(world, x, y - 1);
            }
            else if ((y < World.WORLDSIZE - 1) && (world.getAt(x, y + 1) == null))
            {
                new Anteater(world, x, y + 1);
            }
            else if ((x > 0) && (world.getAt(x - 1, y) == null))
            {
                new Anteater(world, x - 1, y);
            }
            else if ((x < World.WORLDSIZE - 1) && (world.getAt(x + 1, y) == null))
            {
                new Anteater(world, x + 1, y);
            }
        }
    }

    // Basic code reused from Ant.move() method, with optimizations for move checking and additions
    // to handle starvation.

    /**
     * Moves an Anteater object around and handles the starvation counter.
     */
    public void move()
    {
        starveTicks++;
        int direction = (int) (Math.random() * 4);

        // up
        if (direction == 0)
        {
            if ((y > 0) && !(world.getAt(x, y - 1) instanceof Anteater))
            {
                // Reset starvation counter if Anteater "ate".
                if (world.getAt(x, y - 1) instanceof Ant)
                    starveTicks = 0;

                // Move to new spot.
                world.setAt(x, y - 1, world.getAt(x, y));
                world.setAt(x, y, null);
                y--;
            }
        }
        // down
        else if (direction == 1)
        {
            if ((y < World.WORLDSIZE - 1) && !(world.getAt(x, y + 1) instanceof Anteater))
            {
                // Reset starvation counter if Anteater "ate".
                if (world.getAt(x, y + 1) instanceof Ant)
                    starveTicks = 0;

                // Move to new spot.
                world.setAt(x, y + 1, world.getAt(x, y));
                world.setAt(x, y, null);
                y++;
            }
        }
        // left
        else if (direction == 2)
        {
            if ((x > 0) && !(world.getAt(x - 1, y) instanceof Anteater))
            {
                // Reset starvation counter if Anteater "ate".
                if (world.getAt(x - 1, y) instanceof Ant)
                    starveTicks = 0;

                // Move to new spot.
                world.setAt(x - 1, y, world.getAt(x, y));
                world.setAt(x, y, null);
                x--;
```

```java
            }
        }
        // right
        else
        {
            if ((x < World.WORLDSIZE - 1) && !(world.getAt(x + 1, y) instanceof Anteater))
            {
                // Reset starvation counter if Anteater "ate".
                if (world.getAt(x + 1, y) instanceof Ant)
                    starveTicks = 0;

                // Move to new spot.
                world.setAt(x + 1, y, world.getAt(x, y));
                world.setAt(x, y, null);
                x++;
            }
        }
    }

    /**
     * Checks to see if the anteater is starving.
     *
     * @return boolean, indicating whether the anteater is starving (true) or not (false).
     */
    public boolean starve()
    {
        return (starveTicks == ANTEATER_STARVE ? true : false);
    }

    /**
     * Returns "X" as the printable character for an Anteater object.
     */
    public String getPrintableChar()
    {
        return "X";
    }
} // Anteater

```

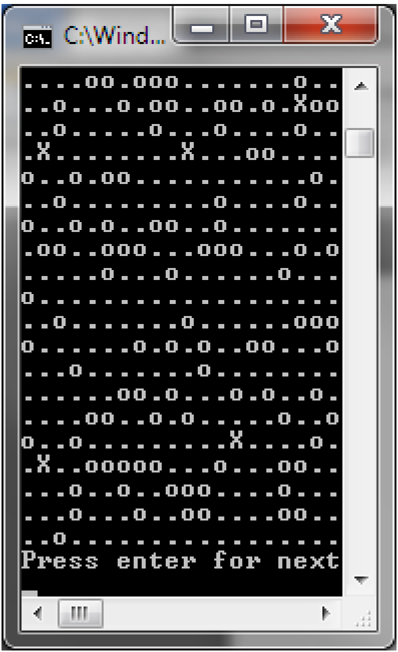
Read from left to right, top to bottom.



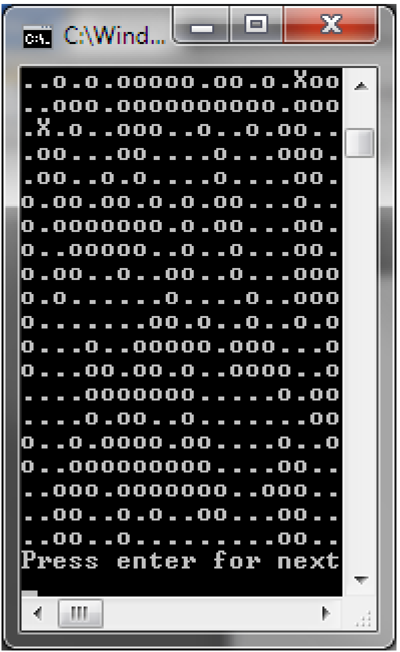
```
....oo.o.o......o.X.
.o.....oo....ooo..oo
.o...oo...oo..o.o..o
ooo...oo..X......o..
X..........o........
.o..o....o.....o....
o.o...oo....o.o.....
...o.....o.oo.......
..o.o.o.........o..o
........oo.....o....
.o..o.....o..o....oo
.......o.o....o..oo.
.o....o...o....o..o.
........o..o...o..o.
o.ooo...o...X.......
....o.o.........o.o.
.X..o.ooo.o.o...oo..
...o.o.o..o....o....
...........o.....o..
..oo....oo.o.....o..
Press enter for next
```



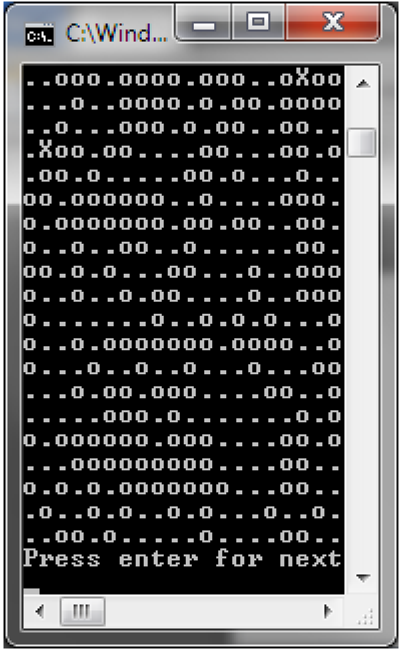
```
....oo.ooo......oX..
..o....o..o..ooo...o
.o....o....o..oo..oo
oo...oo.o..X........
.Xo.........o....o..
................o...
.oo.o....o.oo.......
o.ooo.oo.o....o.....
......o.o...o....o.o
.o........o.....o...
...o.......o.....ooo
.o....o...o..o.o..o.
.......o...o...o..o.
...oo....oo...o....o
o.....o..o..........
.Xo..o......X...o..o
.....ooo.oo..o.o....
....o.o.o..o....oo..
...o....oo.o....oo..
..oo................
Press enter for next
```



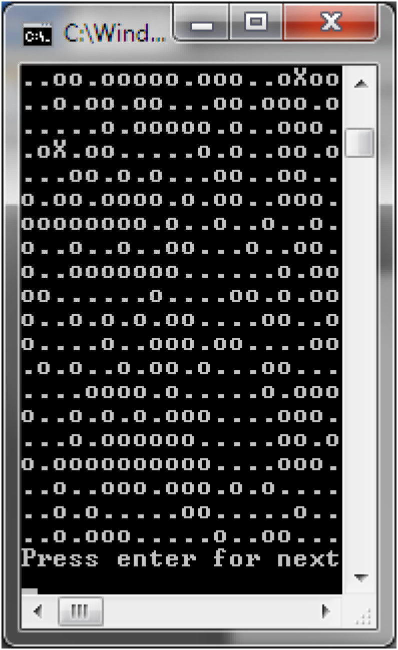
```
....oo.ooo.......o..
..o...o.oo..oo.o.Xoo
..o.....o...o....o..
.X........X...oo....
o..o.oo...........o.
..o.........o....o..
o..o.o..oo..o.......
.oo..ooo...ooo...o.o
.....o...o......o...
o...................
..o.......o......ooo
o......o.o.o..oo...o
...o.......o........
......oo.o...o.o..o.
....oo..o.o.....o..o
o..o.........X....o.
.X..ooooo...o...oo..
...o..o..ooo....o...
...o...o..oo....oo..
..o.................
Press enter for next
```



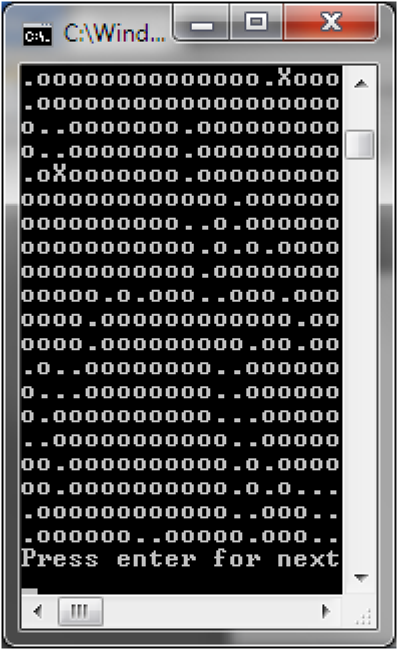
```
..o.o.ooooo.oo.o.Xoo
..ooo.oooooooooo.ooo
.X.o..ooo..o..o.oo..
.oo...oo....o...ooo.
.oo..o.o....o....oo.
o.oo.oo.o.o.oo...o..
o.ooooooo.o.oo...oo.
o..ooooo..o..o...oo.
o.oo..o..oo..o...ooo
o.o......o....o..ooo
o.......oo.o..o..o.o
o...o..ooooo.ooo...o
o...oo.oo.o..oooo..o
....ooooooo.....o.oo
....o.oo..o.......oo
o..o.oooo.oo....o..o
o..ooooooooo....oo..
..ooo.ooooooo..ooo..
..oo..o.o..oo...oo..
..oo..o.........oo..
Press enter for next
```



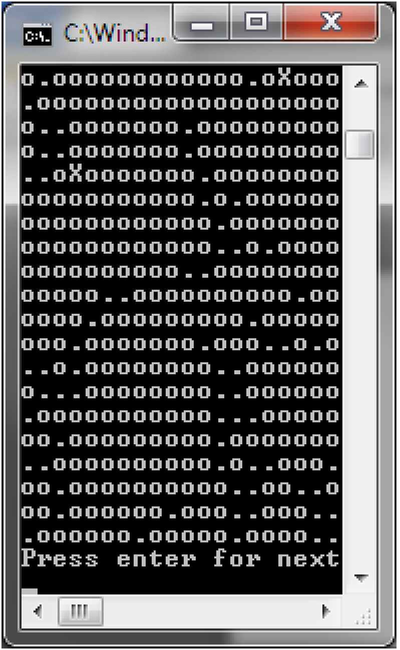
```
..ooo.oooo.ooo..oXoo
...o..oooo.o.oo.oooo
..o...ooo.o.oo..oo..
.Xoo.oo....oo...oo.o
.oo.o.....oo.o...o..
oo.oooooo..o....ooo.
o.ooooooo.oo.oo..oo.
o..o..oo..o......oo.
oo.o.o...oo...o..ooo
o..o..o.oo....o..ooo
o.......o..o.o.o...o
o..o.ooooooo.oooo..o
o...o..o..o...o...oo
...o.oo.ooo....oo..o
.....ooo.o.......o.o
o.oooooo.ooo....oo.o
...ooooooooo....oo..
o.o.o.ooooooo...oo..
.o..o.o..o.o...o..o.
..oo.o.....o....oo..
Press enter for next
```



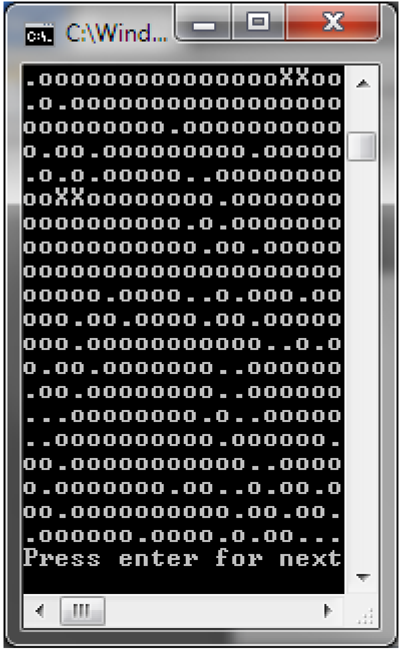
```
..oo.ooooo.ooo..oXoo
..o.oo.oo...oo.ooo.o
.....o.ooooo.o..ooo.
.oX.oo.....o.o..oo.o
...oo.o.o...oo..oo..
o.oo.oooo.o.oo..ooo.
oooooooo.o..o..o..o.
o..o..o..oo...o..oo.
o..ooooooo......o.oo
oo......o....oo.o.oo
o..o.o.o.oo....oo..o
o....o..ooo.oo....oo
.o.o..o.oo.o...oo...
....oooo.o.....o.ooo
o..o.o.o.ooo....ooo.
...o.oooooo.....oo.o
o.oooooooooo....ooo.
..o..ooo.ooo.o.o....
..o.o.....oo.....o..
..o.ooo.....o..oo...
Press enter for next
```



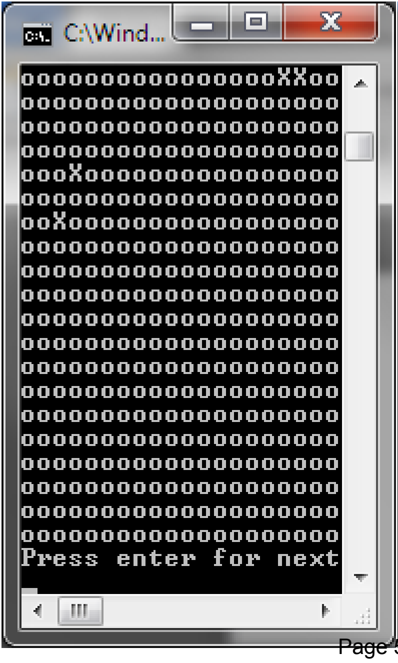
```
.oooooooooooooo.Xooo
.ooooooooooooooooooo
o..ooooooo.ooooooooo
o..ooooooo.ooooooooo
.oXooooooo.ooooooooo
ooooooooooooo.oooooo
oooooooooo..o.oooooo
ooooooooooo.o.o.oooo
ooooooooooo.oooooooo
ooooo.o.ooo..ooo.ooo
oooo.oooooooooooo.oo
oooo.ooooooooo.oo.oo
.o..oooooooo..oooooo
o...oooooooo..oooooo
o.oooooooooo...ooooo
..ooooooooooo..ooooo
oo.oooooooooo.o.oooo
oo.oooooooooo.o.o...
.oooooooooooo..ooo..
.oooooo..ooooo.ooo..
Press enter for next
```



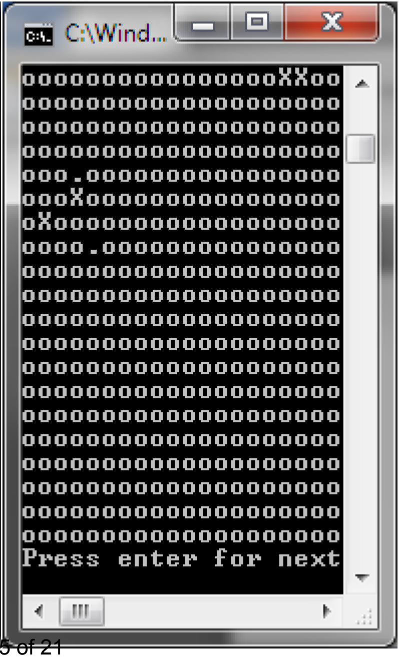
```
o.oooooooooooo.oXooo
.ooooooooooooooooooo
o..ooooooo.ooooooooo
o..ooooooo.ooooooooo
..oXooooooo.oooooooo
ooooooooooo.o.oooooo
oooooooooooo.ooooooo
oooooooooooo..o.oooo
oooooooooo..oooooooo
ooooo..oooooooooo.oo
oooo.ooooooooo.ooooo
ooo.ooooooo.ooo..o.o
..o.oooooooo..oooooo
o...oooooooo..oooooo
.ooooooooooo...ooooo
oo.ooooooooo.ooooooo
..oooooooooo.o..ooo.
oo.oooooooooo..oo..o
oo.oooooo.ooo..ooo..
.oooooo.ooooo.oooo..
Press enter for next
```



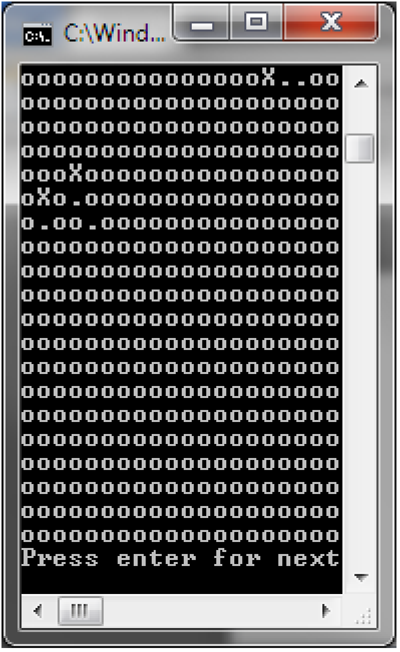
```
.oooooooooooooooXXoo
.o.ooooooooooooooooo
ooooooooo.oooooooooo
o.oo.ooooooooo.ooooo
.o.o.ooooo..oooooooo
ooXXoooooooo.ooooooo
oooooooooo.o.ooooooo
ooooooooooo.oo.ooooo
oooooooooooooooooooo
ooooo.oooo..o.ooo.oo
ooo.oo.oooo.oo.ooooo
ooo.ooooooooooo..o.o
o.oo.ooooooo..oooooo
.oo.oooooooo..oooooo
...oooooooo.o..ooooo
..oooooooooo.oooooo.
oo.ooooooooooo..oooo
o.ooooooo.oo..o.oo.o
oo.oooooooooo.oo.oo.
.oooooo.oooo.o.oo...
Press enter for next
```



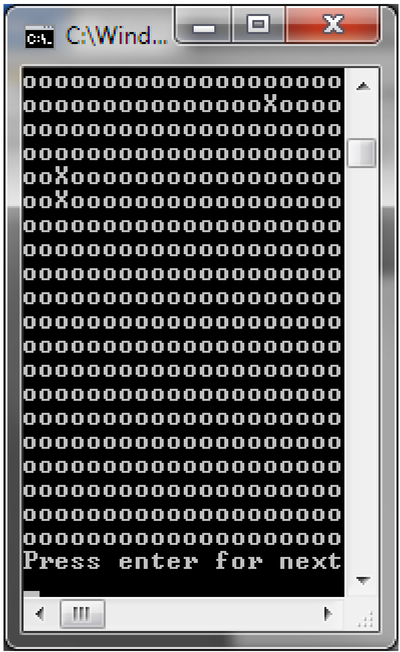
```
ooooooooooooooooXXoo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooXoooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



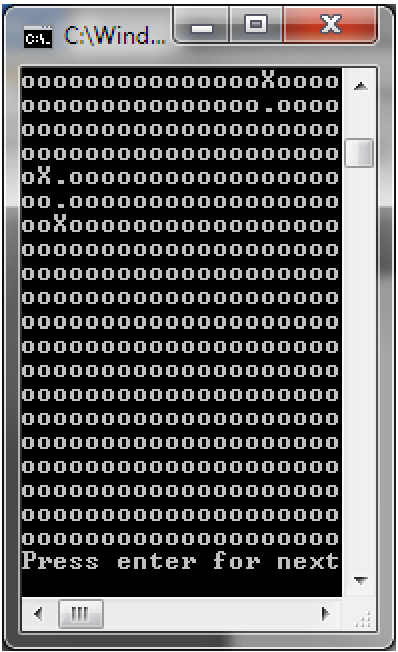
```
ooooooooooooooooXXoo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooo.oooooooooooooooo
oooXoooooooooooooooo
oXoooooooooooooooooo
oooo.ooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



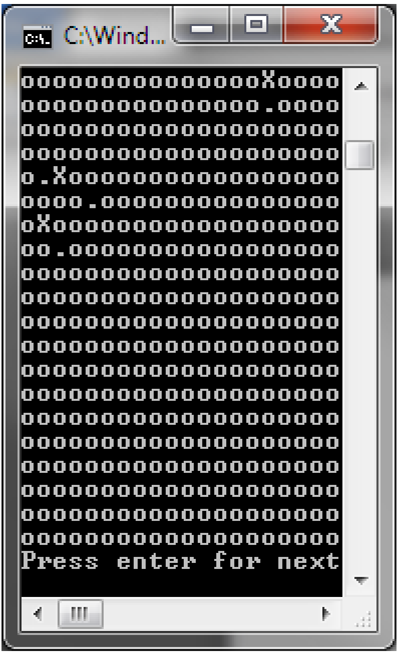
```
oooooooooooooooX..oo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooXoooooooooooooooo
oXo.oooooooooooooooo
o.oo.ooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



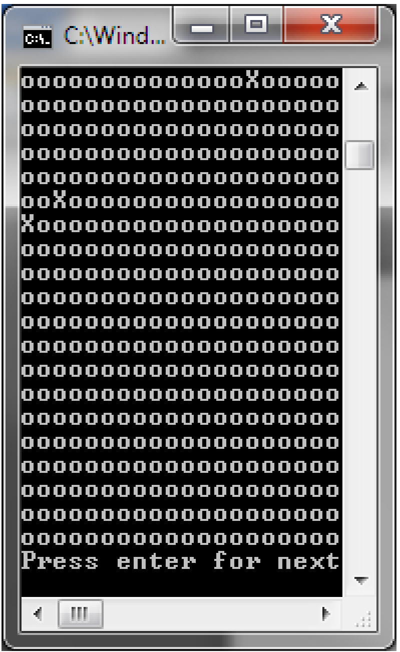
```
oooooooooooooooooooo
oooooooooooooooXoooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
ooXooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



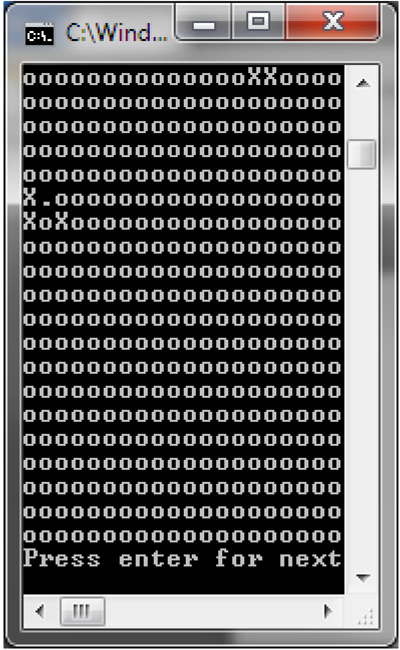
```
oooooooooooooooXoooo
ooooooooooooooo.oooo
oooooooooooooooooooo
oooooooooooooooooooo
oX.ooooooooooooooooo
oo.ooooooooooooooooo
ooXooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



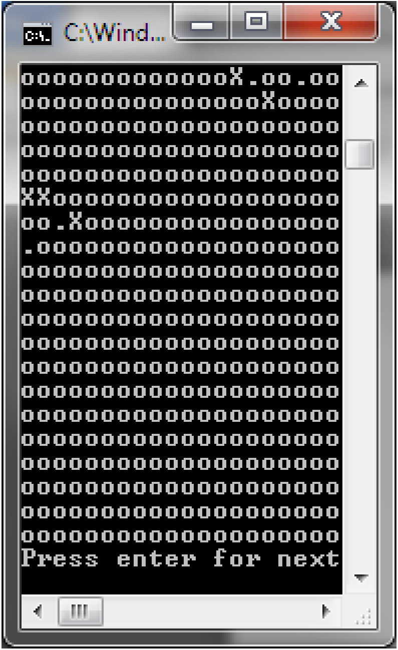
```
oooooooooooooooXoooo
ooooooooooooooo.oooo
oooooooooooooooooooo
oooooooooooooooooooo
o.Xooooooooooooooooo
oooo.ooooooooooooooo
oXoooooooooooooooooo
oo.ooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



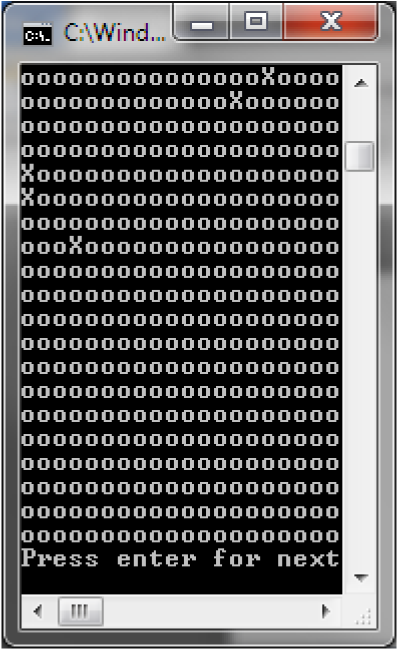
```
ooooooooooooooXooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
Xooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



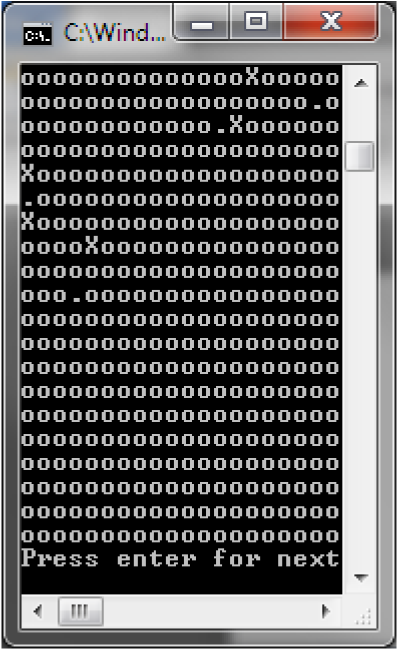
```
ooooooooooooooXXoooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
X.oooooooooooooooooo
XoXooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



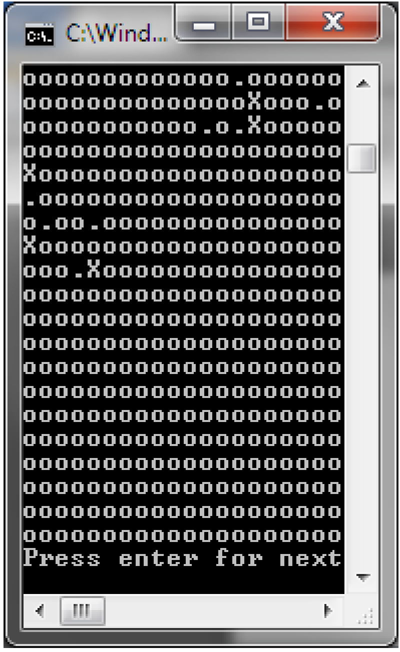
```
oooooooooooooX.oo.oo
oooooooooooooooXoooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
XXoooooooooooooooooo
oo.Xoooooooooooooooo
.ooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



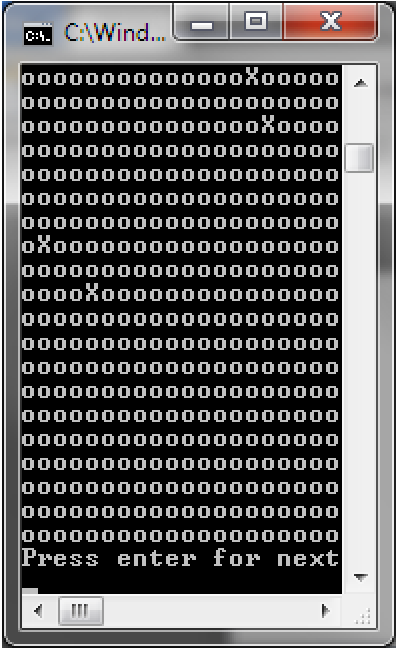
```
oooooooooooooooXoooo
oooooooooooooXoooooo
oooooooooooooooooooo
oooooooooooooooooooo
Xooooooooooooooooooo
Xooooooooooooooooooo
oooooooooooooooooooo
oooXoooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



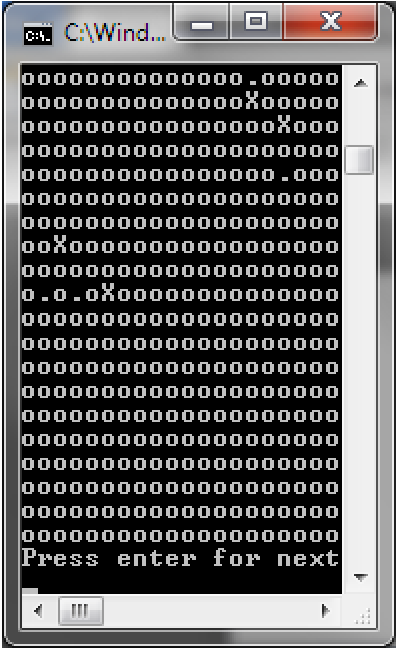
```
ooooooooooooooXooooo
oooooooooooooooooo.o
oooooooooooo.Xoooooo
oooooooooooooooooooo
Xooooooooooooooooooo
.ooooooooooooooooooo
Xooooooooooooooooooo
ooooXooooooooooooooo
oooooooooooooooooooo
ooo.oooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



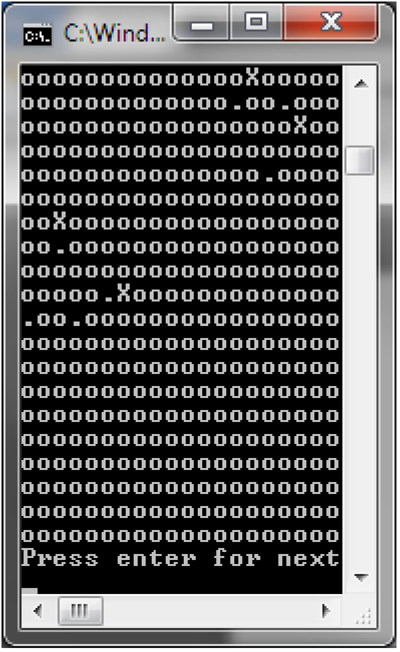
```
ooooooooooooo.oooooo
ooooooooooooooXooo.o
ooooooooooo.o.Xooooo
oooooooooooooooooooo
Xooooooooooooooooooo
.ooooooooooooooooooo
o.oo.ooooooooooooooo
Xooooooooooooooooooo
ooo.Xooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



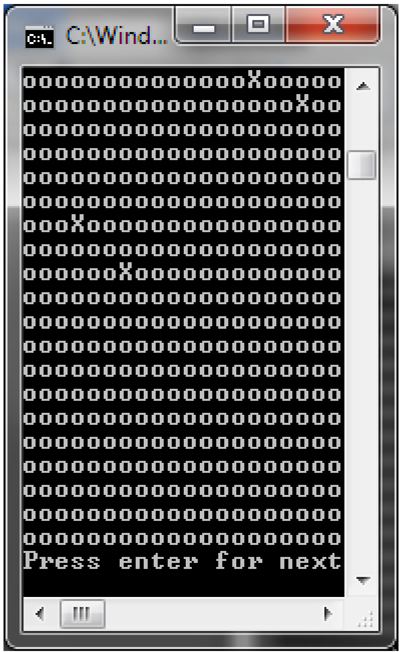
```
ooooooooooooooXooooo
oooooooooooooooooooo
oooooooooooooooXoooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oXoooooooooooooooooo
oooooooooooooooooooo
ooooXooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



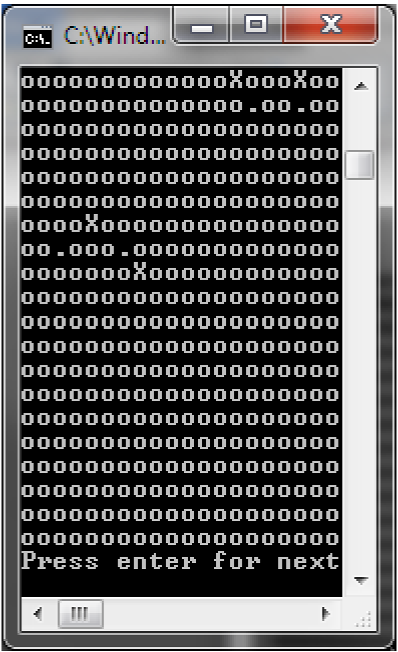
```
oooooooooooooo.ooooo
ooooooooooooooXooooo
ooooooooooooooooXooo
oooooooooooooooooooo
oooooooooooooooo.ooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
oooooooooooooooooooo
o.o.oXoooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



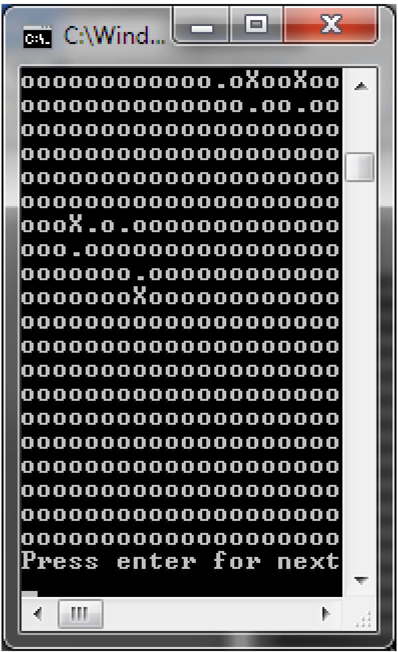
```
ooooooooooooooXooooo
ooooooooooooo.oo.ooo
oooooooooooooooooXoo
oooooooooooooooooooo
ooooooooooooooo.oooo
oooooooooooooooooooo
ooXooooooooooooooooo
oo.ooooooooooooooooo
oooooooooooooooooooo
ooooo.Xooooooooooooo
.oo.oooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



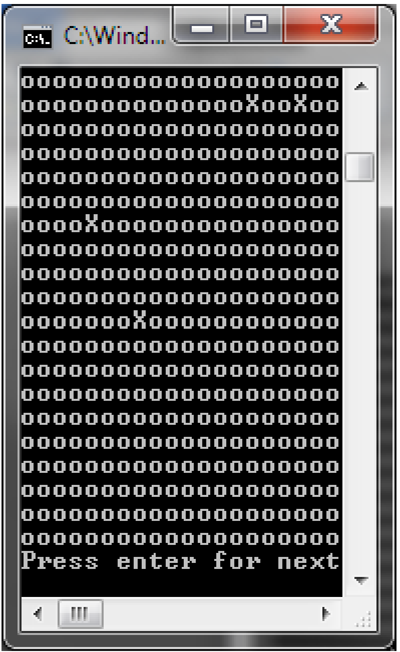
```
ooooooooooooooXooooo
oooooooooooooooooXoo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooXoooooooooooooooo
oooooooooooooooooooo
ooooooXooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



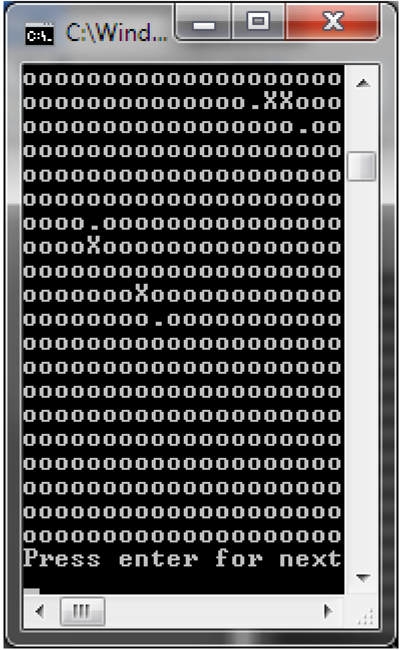
```
oooooooooooooXoooXoo
oooooooooooooo.oo.oo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooooXooooooooooooooo
oo.ooo.ooooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



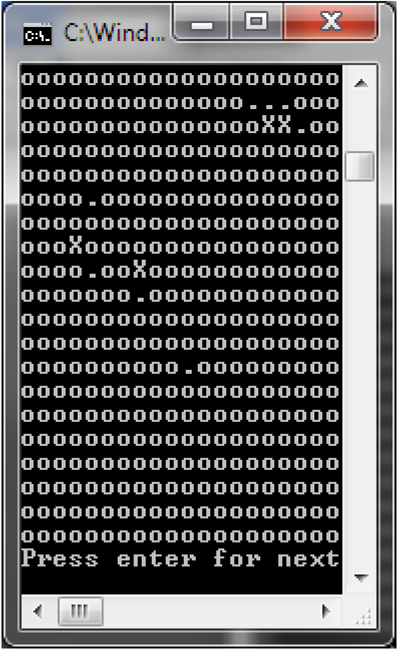
```
oooooooooooo.oXooXoo
oooooooooooooo.oo.oo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooX.o.ooooooooooooo
ooo.oooooooooooooooo
ooooooo.oooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



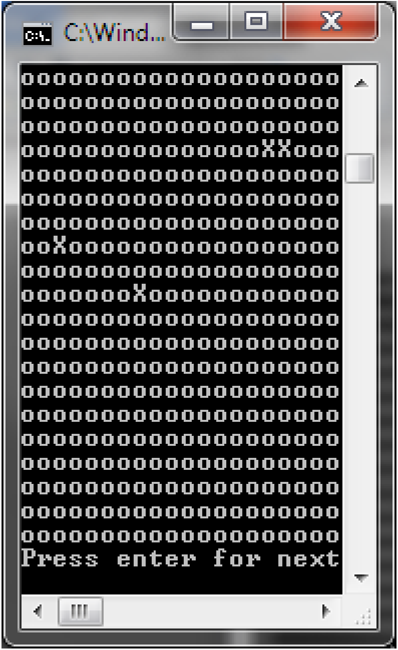
```
oooooooooooooooooooo
ooooooooooooooXooXoo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooooXooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



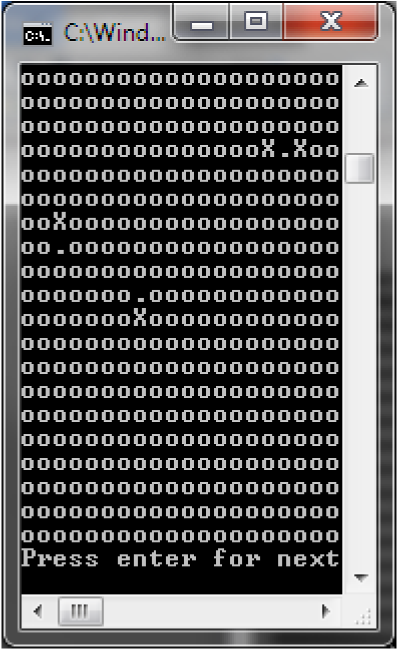
```
oooooooooooooooooooo
oooooooooooooo.XXooo
ooooooooooooooooo.oo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooo.ooooooooooooooo
ooooXooooooooooooooo
oooooooooooooooooooo
oooooooXoooooooooooo
oooooooo.ooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



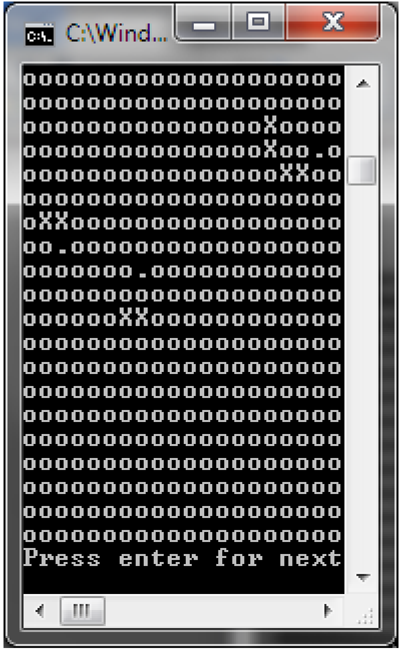
```
oooooooooooooooooooo
oooooooooooooo...ooo
oooooooooooooooXX.oo
oooooooooooooooooooo
oooooooooooooooooooo
oooo.ooooooooooooooo
oooooooooooooooooooo
oooXoooooooooooooooo
oooo.ooXoooooooooooo
ooooooo.oooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooo.ooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



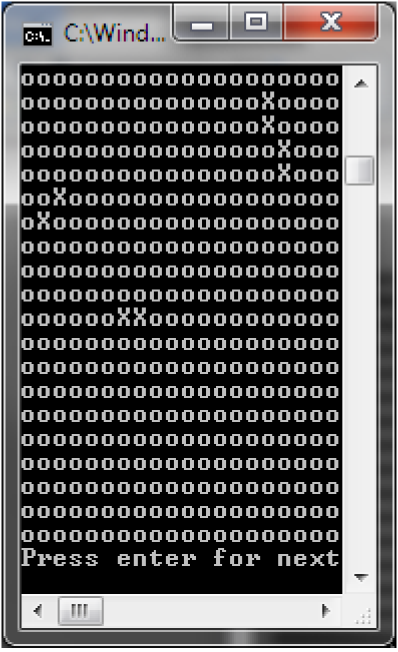
```
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooXXooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
oooooooooooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



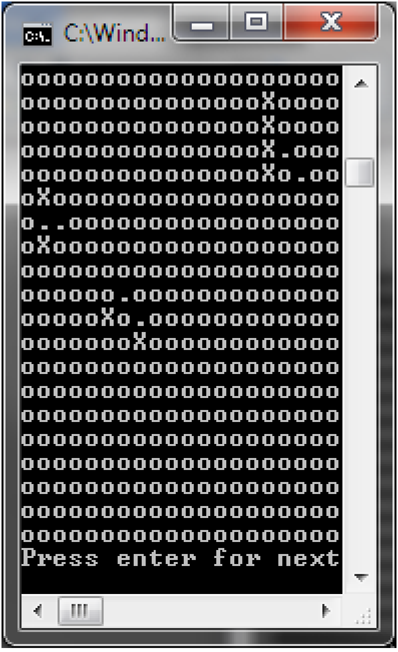
```
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooX.Xoo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
oo.ooooooooooooooooo
oooooooooooooooooooo
ooooooo.oooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



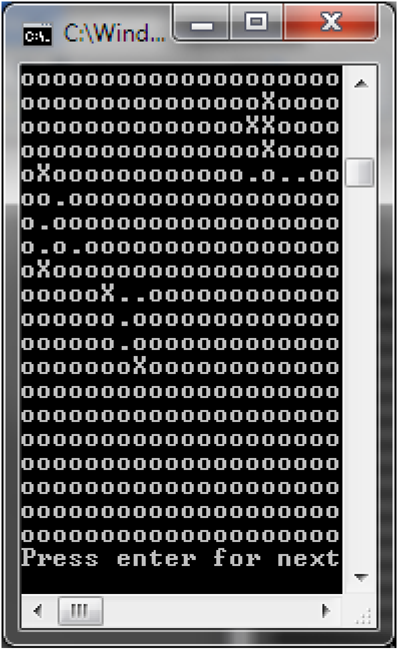
```
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooXoooo
oooooooooooooooXoo.o
ooooooooooooooooXXoo
oooooooooooooooooooo
oXXooooooooooooooooo
oo.ooooooooooooooooo
ooooooo.oooooooooooo
oooooooooooooooooooo
ooooooXXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



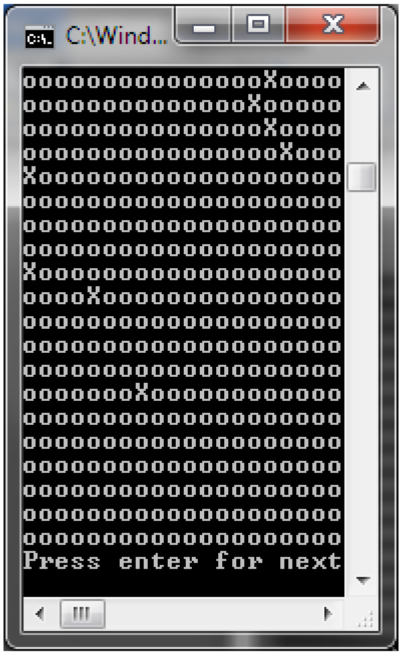
```
oooooooooooooooooooo
oooooooooooooooXoooo
oooooooooooooooXoooo
ooooooooooooooooXooo
ooooooooooooooooXooo
ooXooooooooooooooooo
oXoooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooooooXXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



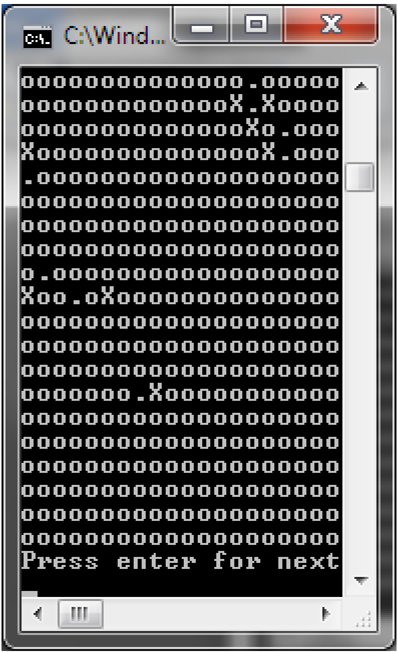
```
oooooooooooooooooooo
oooooooooooooooXoooo
oooooooooooooooXoooo
oooooooooooooooX.ooo
oooooooooooooooXo.oo
oXoooooooooooooooooo
o..ooooooooooooooooo
oXoooooooooooooooooo
oooooooooooooooooooo
oooooo.ooooooooooooo
oooooXo.oooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



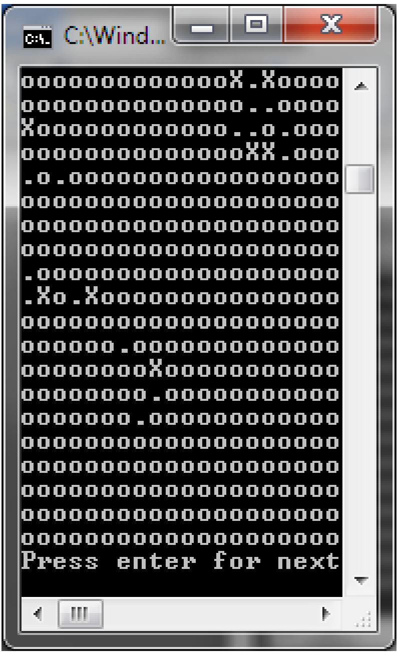
```
oooooooooooooooooooo
oooooooooooooooXoooo
ooooooooooooooXXoooo
oooooooooooooooXoooo
oXoooooooooooo.o..oo
oo.ooooooooooooooooo
o.oooooooooooooooooo
o.o.oooooooooooooooo
oXoooooooooooooooooo
oooooX..oooooooooooo
oooooo.ooooooooooooo
oooooo.ooooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



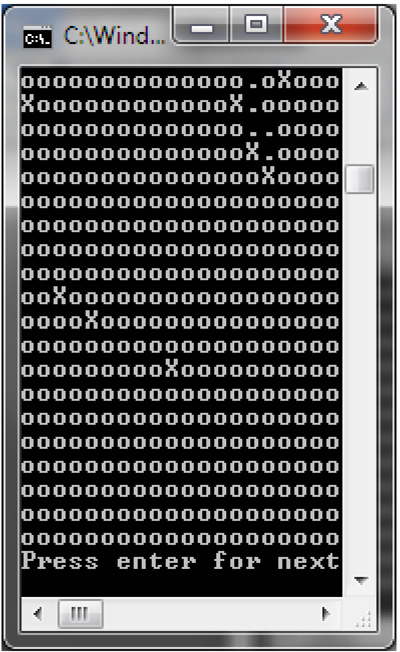
```
oooooooooooooooXoooo
ooooooooooooooXooooo
oooooooooooooooXoooo
ooooooooooooooooXooo
Xooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Xooooooooooooooooooo
ooooXooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



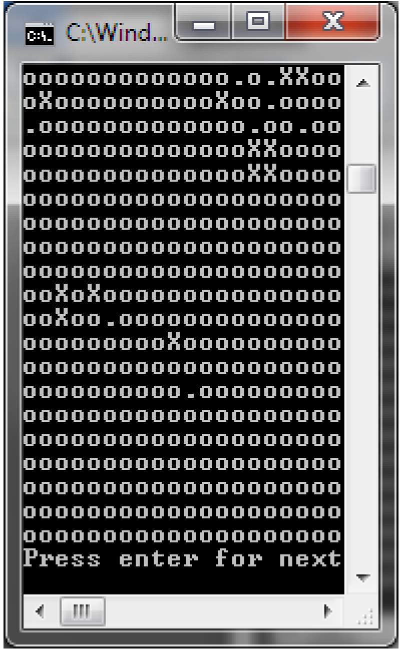
```
oooooooooooooo.ooooo
oooooooooooooX.Xoooo
ooooooooooooooXo.ooo
XooooooooooooooX.ooo
.ooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
o.oooooooooooooooooo
Xoo.oXoooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooooooo.Xooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



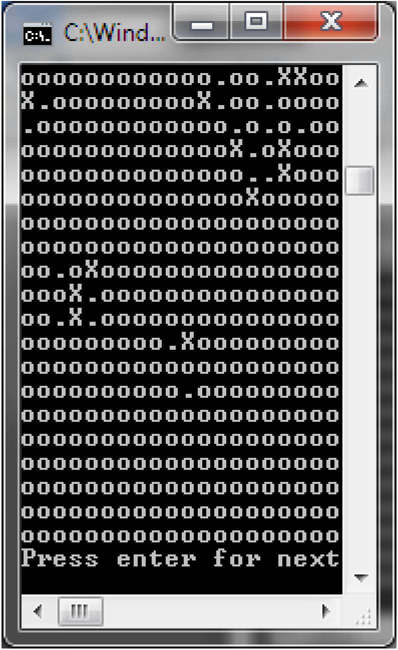
```
oooooooooooooX.Xoooo
oooooooooooooo..oooo
Xoooooooooooo..o.ooo
ooooooooooooooXX.ooo
.o.ooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
.ooooooooooooooooooo
.Xo.Xooooooooooooooo
oooooooooooooooooooo
oooooo.ooooooooooooo
ooooooooXooooooooooo
oooooooo.ooooooooooo
ooooooo.oooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



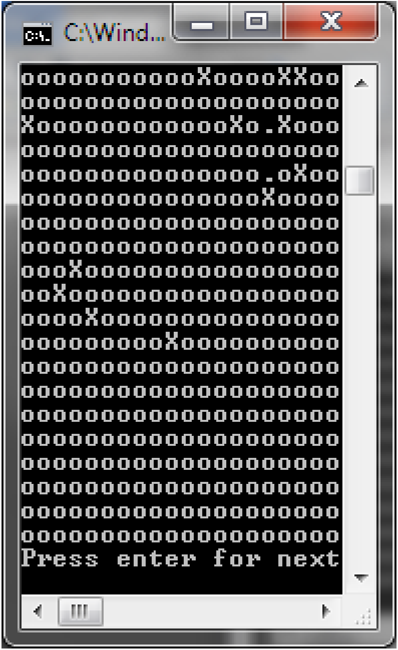
```
oooooooooooooo.oXooo
XooooooooooooX.ooooo
oooooooooooooo..oooo
ooooooooooooooX.oooo
oooooooooooooooXoooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
ooooXooooooooooooooo
oooooooooooooooooooo
oooooooooXoooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



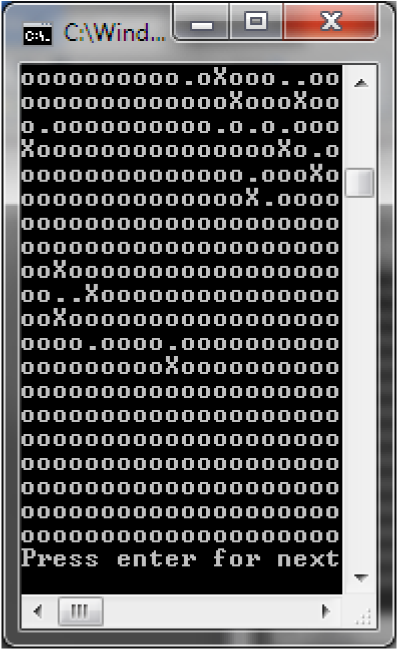
```
ooooooooooooo.o.XXoo
oXooooooooooXoo.oooo
.ooooooooooooo.oo.oo
ooooooooooooooXXoooo
ooooooooooooooXXoooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXoXooooooooooooooo
ooXoo.oooooooooooooo
oooooooooXoooooooooo
oooooooooooooooooooo
oooooooooo.ooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



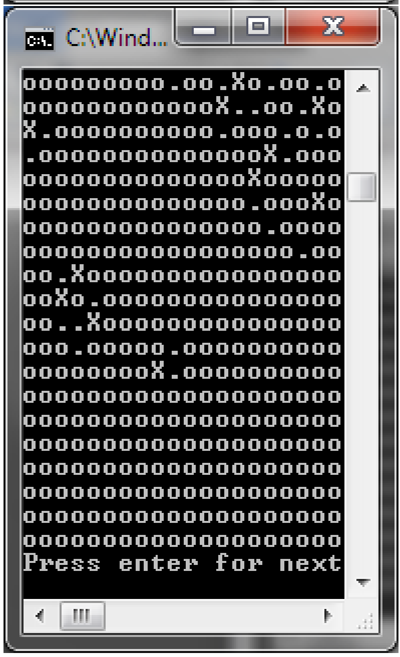
```
oooooooooooo.oo.XXoo
X.oooooooooX.oo.oooo
.oooooooooooo.o.o.oo
oooooooooooooX.oXooo
oooooooooooooo..Xooo
ooooooooooooooXooooo
oooooooooooooooooooo
oooooooooooooooooooo
oo.oXooooooooooooooo
oooX.ooooooooooooooo
oo.X.ooooooooooooooo
ooooooooo.Xooooooooo
oooooooooooooooooooo
oooooooooo.ooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



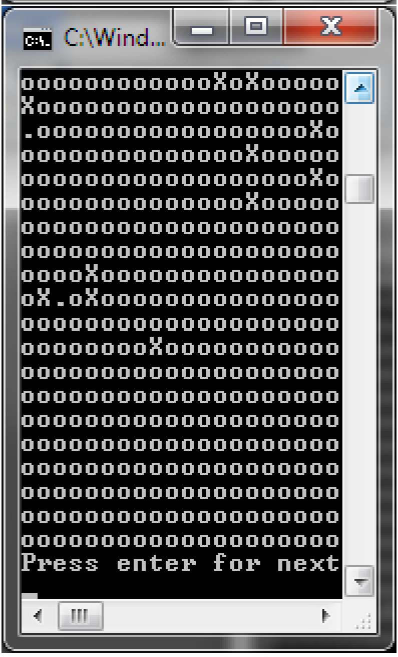
```
oooooooooooXooooXXoo
oooooooooooooooooooo
XooooooooooooXo.Xooo
oooooooooooooooooooo
ooooooooooooooo.oXoo
oooooooooooooooXoooo
oooooooooooooooooooo
oooooooooooooooooooo
oooXoooooooooooooooo
ooXooooooooooooooooo
ooooXooooooooooooooo
oooooooooXoooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



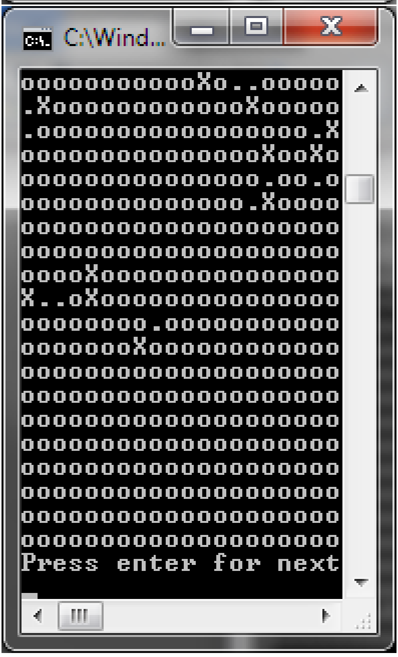
```
oooooooooo.oXooo..oo
oooooooooooooXoooXoo
o.oooooooooo.o.o.ooo
XoooooooooooooooXo.o
oooooooooooooo.oooXo
ooooooooooooooX.oooo
oooooooooooooooooooo
oooooooooooooooooooo
ooXooooooooooooooooo
oo..Xooooooooooooooo
ooXooooooooooooooooo
oooo.oooo.oooooooooo
oooooooooXoooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



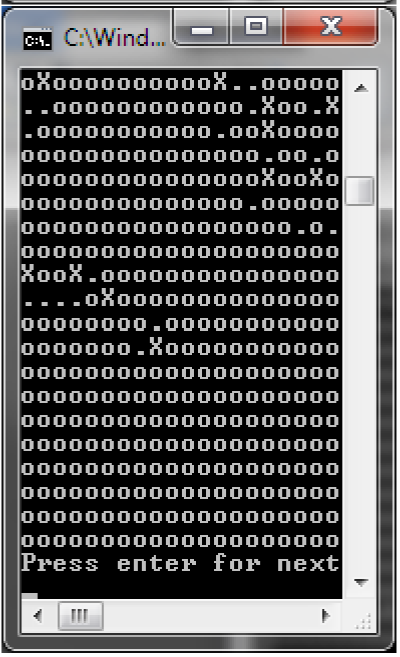
```
ooooooooo.oo.Xo.oo.o
ooooooooooooX..oo.Xo
X.oooooooooo.ooo.o.o
.ooooooooooooooX.ooo
ooooooooooooooXooooo
oooooooooooooo.oooXo
ooooooooooooooo.oooo
ooooooooooooooooo.oo
oo.Xoooooooooooooooo
ooXo.ooooooooooooooo
oo..Xooooooooooooooo
ooo.ooooo.oooooooooo
ooooooooX.oooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



```
ooooooooooooXoXooooo
Xooooooooooooooooooo
.oooooooooooooooooXo
ooooooooooooooXooooo
ooooooooooooooooooXo
ooooooooooooooXooooo
oooooooooooooooooooo
oooooooooooooooooooo
ooooXooooooooooooooo
oX.oXooooooooooooooo
oooooooooooooooooooo
ooooooooXooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



```
oooooooooooXo..ooooo
.XooooooooooooXooooo
.ooooooooooooooooo.X
oooooooooooooooXooXo
ooooooooooooooo.oo.o
oooooooooooooo.Xoooo
oooooooooooooooooooo
oooooooooooooooooooo
ooooXooooooooooooooo
X..oXooooooooooooooo
oooooooo.ooooooooooo
oooooooXoooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```



```
oXooooooooooX..ooooo
..oooooooooooo.Xoo.X
.ooooooooooo.ooXoooo
ooooooooooooooo.oo.o
oooooooooooooooXooXo
oooooooooooooo.ooooo
ooooooooooooooooo.o.
oooooooooooooooooooo
XooX.ooooooooooooooo
....oXoooooooooooooo
oooooooo.ooooooooooo
ooooooo.Xooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
oooooooooooooooooooo
Press enter for next
```

Name: Dan Cassidy

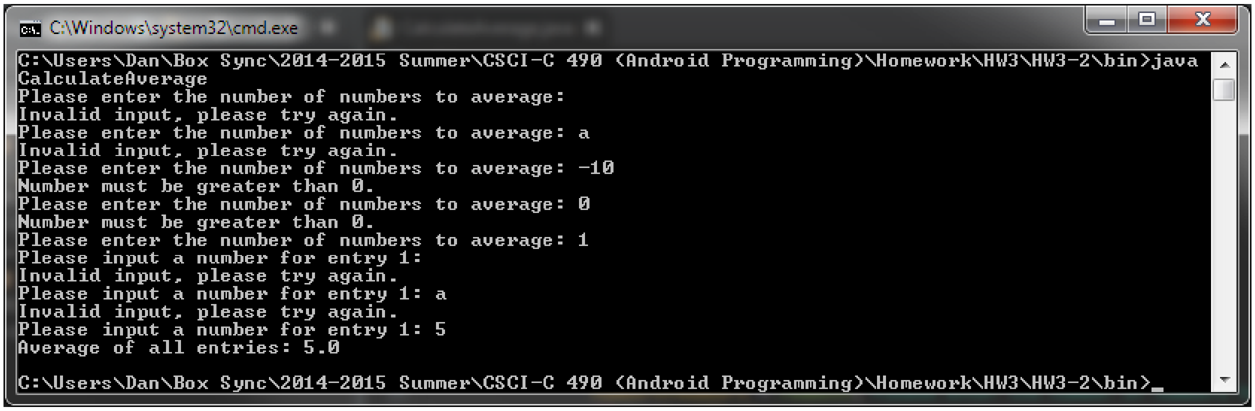
Class: CSCI-C 490, Mobile Application Development

Assignment: Homework 3 Part 2

Date: 2015-07-10

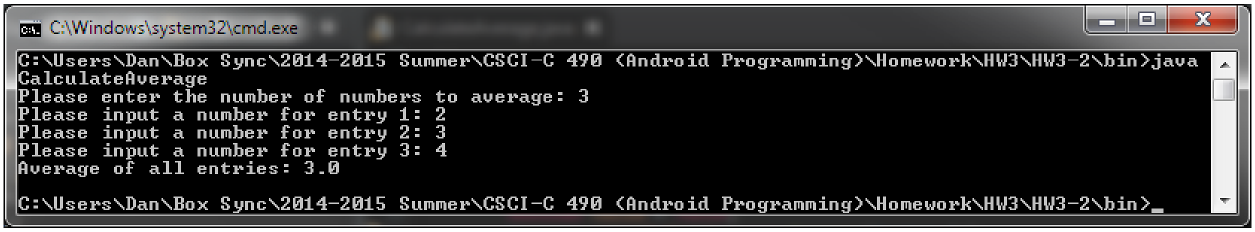
```java
/*--------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-07-10
 * Assignment:  HW3-2
 * Source File: CalculateAverage.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
--------------------------------------------------------------------------------------------------*/
import java.util.Scanner;

/**
 * A small class to calculate the average of a given number of integers.
 *
 * @author Dan Cassidy
 */
public class CalculateAverage
{

    public static void main(String[] args)
    {
        int numberOfNumbers = 0;
        boolean valid = false;
        Scanner consoleInput = new Scanner(System.in);

        // Loop while input is not valid.
        while (!valid)
        {
            try
            {
                numberOfNumbers = readInt("Please enter the number of numbers to average: ");
                if (numberOfNumbers <= 0)
                    throw new Exception("Number must be greater than 0.");
                valid = true;
            }
            catch (Exception ex)
            {
                System.out.println(ex.getMessage());
            }
        }

        // Declare an array of the specified size and then ask for input for all elements.
        int[] numbers = new int[numberOfNumbers];
        for (int counter = 0; counter < numbers.length; counter++)
            numbers[counter] = readInt("Please input a number for entry " + (counter + 1) + ": ");

        System.out.println("Average of all entries: " + average(numbers));
    }

    /**
     * Computes the average (arithmetic mean) of an array of numbers. If <b>numbers</b> is null or
     * an empty array, 0 is returned.
     *
     * @param numbers An array of integers, from which their average will be computed.
     * @return double, representing the average of the elements contained in <b>numbers</b>.
     */
    public static double average(int[] numbers)
    {
        // No need to throw an error, just return 0 if the argument is no good.
        if (numbers == null || numbers.length == 0)
            return 0;
        // Compute and return the average.
        double sum = 0;
        for (int number : numbers)
            sum += number;
        return sum / numbers.length;
    }

    /**
     * Reads an integer from the console.
     *
     * @param prompt A String object containing the prompt text for a user entering a number.
     * @return int, holding the integer read from the console.
     */
    public static int readInt(String prompt)
    {
        int number = 0;
        boolean valid = false;
        Scanner consoleInput = new Scanner(System.in);

        // Loop while input is not valid.
        while (!valid)
        {
            try
            {
                System.out.print(prompt);
                number = Integer.parseInt(consoleInput.nextLine());
                valid = true;
            }
            catch (NumberFormatException ex)
            {
                System.out.println("Invalid input, please try again.");
            }
        }

        return number;
    }
}
```

Shows the program handling invalid input, then the edge case of 1 number.



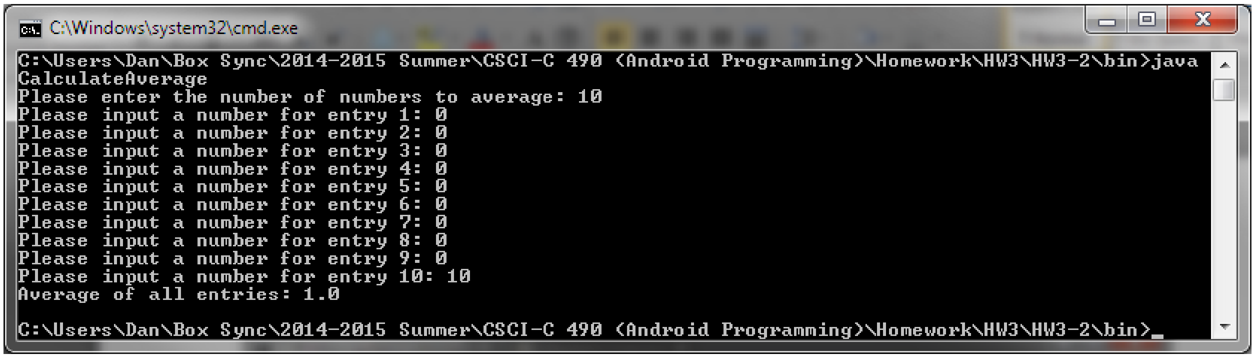
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW3\HW3-2\bin>java
CalculateAverage
Please enter the number of numbers to average:
Invalid input, please try again.
Please enter the number of numbers to average: a
Invalid input, please try again.
Please enter the number of numbers to average: -10
Number must be greater than 0.
Please enter the number of numbers to average: 0
Number must be greater than 0.
Please enter the number of numbers to average: 1
Please input a number for entry 1:
Invalid input, please try again.
Please input a number for entry 1: a
Invalid input, please try again.
Please input a number for entry 1: 5
Average of all entries: 5.0

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW3\HW3-2\bin>_
```

Shows handling 3 numbers.



```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW3\HW3-2\bin>java
CalculateAverage
Please enter the number of numbers to average: 3
Please input a number for entry 1: 2
Please input a number for entry 2: 3
Please input a number for entry 3: 4
Average of all entries: 3.0

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW3\HW3-2\bin>_
```

Shows handling 10 numbers.



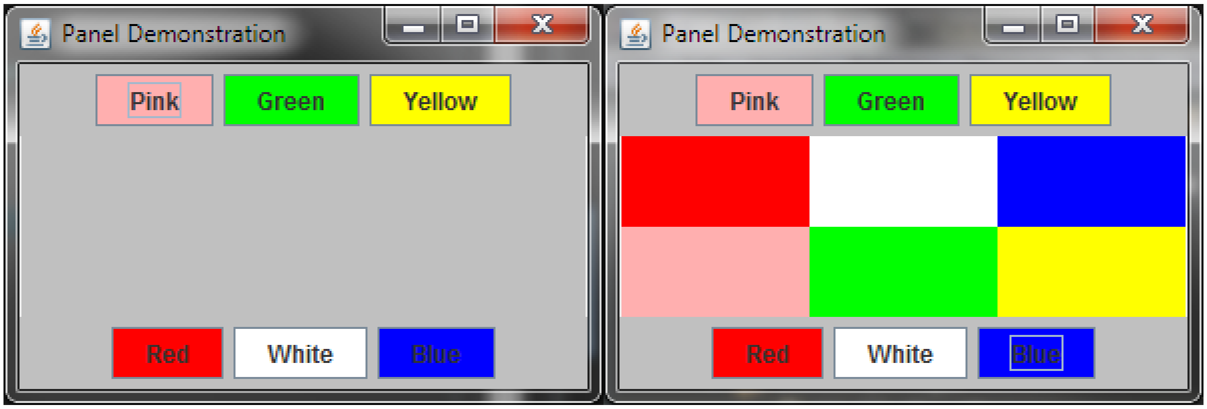
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW3\HW3-2\bin>java
CalculateAverage
Please enter the number of numbers to average: 10
Please input a number for entry 1: 0
Please input a number for entry 2: 0
Please input a number for entry 3: 0
Please input a number for entry 4: 0
Please input a number for entry 5: 0
Please input a number for entry 6: 0
Please input a number for entry 7: 0
Please input a number for entry 8: 0
Please input a number for entry 9: 0
Please input a number for entry 10: 10
Average of all entries: 1.0

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW3\HW3-2\bin>_
```

Name:	Dan Cassidy
Class:	CSCI-C 490, Mobile Application Development
Assignment:	Homework 3 Part 3
Date:	2015-07-13

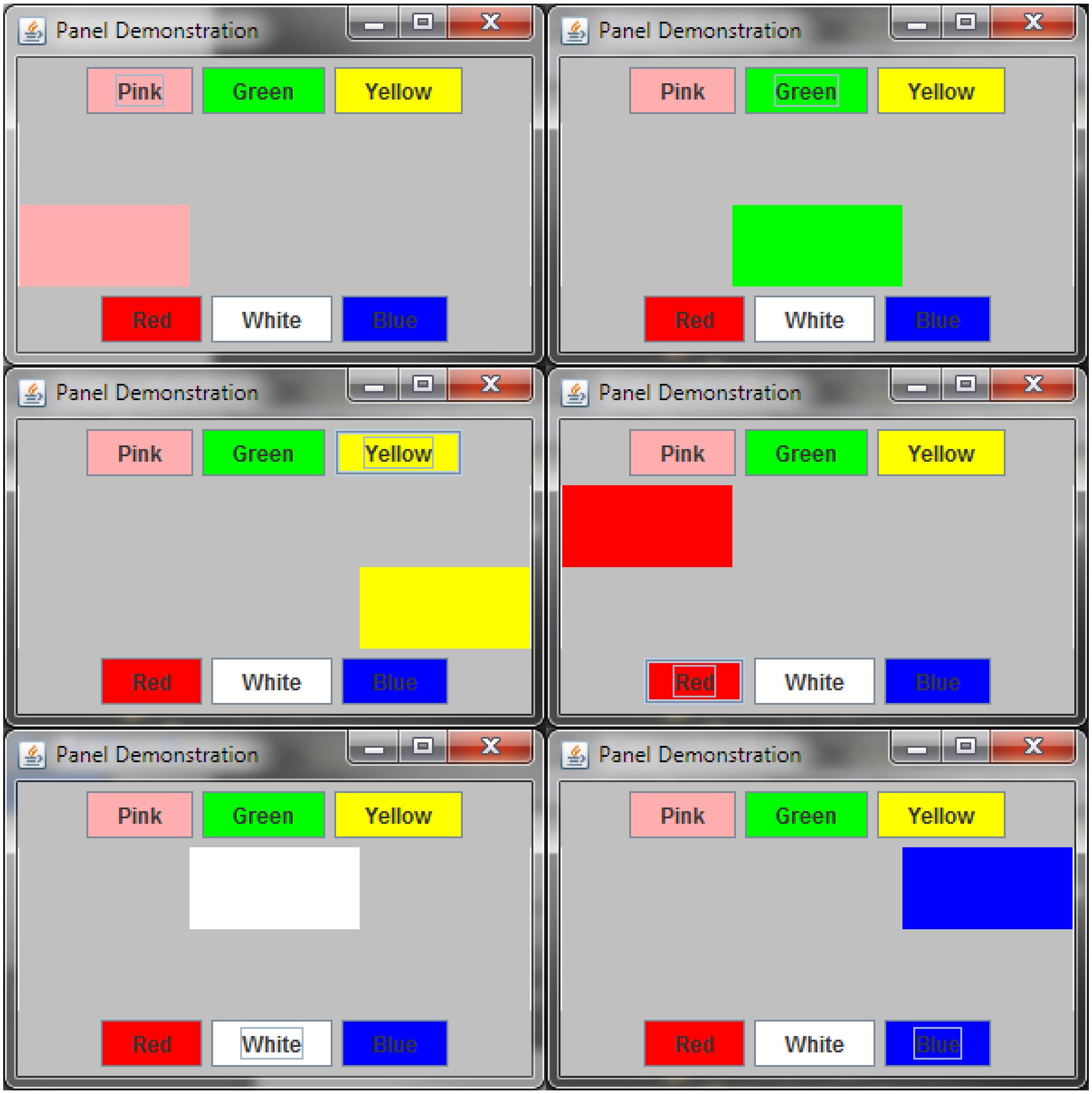
```java
/**
 * Default constructor. Handles the setup of all the GUI elements.
 */
public ModifiedPanelDemo()
{
    super("Panel Demonstration");
    setSize(WIDTH, HEIGHT);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    // Set up the top row of buttons.
    JPanel buttonPanel1 = new JPanel();
    buttonPanel1.setBackground(Color.LIGHT_GRAY);
    buttonPanel1.setLayout(new FlowLayout());

    JButton pinkButton = new JButton("Pink");
    pinkButton.setBackground(Color.PINK);
    pinkButton.addActionListener(this);
    buttonPanel1.add(pinkButton);

    JButton greenButton = new JButton("Green");
    greenButton.setBackground(Color.GREEN);
    greenButton.addActionListener(this);
    buttonPanel1.add(greenButton);

    JButton yellowButton = new JButton("Yellow");
    yellowButton.setBackground(Color.YELLOW);
    yellowButton.addActionListener(this);
    buttonPanel1.add(yellowButton);

    add(buttonPanel1, BorderLayout.NORTH);

    // Set up the main color panel.
    JPanel biggerPanel = new JPanel();
    biggerPanel.setLayout(new GridLayout(2, 3));

    // Begin new row in main color panel.
    redPanel = new JPanel();
    redPanel.setBackground(Color.LIGHT_GRAY);
    biggerPanel.add(redPanel);

    whitePanel = new JPanel();
    whitePanel.setBackground(Color.LIGHT_GRAY);
    biggerPanel.add(whitePanel);

    bluePanel = new JPanel();
    bluePanel.setBackground(Color.LIGHT_GRAY);
    biggerPanel.add(bluePanel);

    // Begin new row in main color panel.
    pinkPanel = new JPanel();
    pinkPanel.setBackground(Color.LIGHT_GRAY);
    biggerPanel.add(pinkPanel);

    greenPanel = new JPanel();
    greenPanel.setBackground(Color.LIGHT_GRAY);
    biggerPanel.add(greenPanel);

    yellowPanel = new JPanel();
    yellowPanel.setBackground(Color.LIGHT_GRAY);
```

```java
        biggerPanel.add(yellowPanel);

        add(biggerPanel, BorderLayout.CENTER);

        // Set up the bottom row of buttons.
        JPanel buttonPanel2 = new JPanel();
        buttonPanel2.setBackground(Color.LIGHT_GRAY);
        buttonPanel2.setLayout(new FlowLayout());

        JButton redButton = new JButton("Red");
        redButton.setBackground(Color.RED);
        redButton.addActionListener(this);
        buttonPanel2.add(redButton);

        JButton whiteButton = new JButton("White");
        whiteButton.setBackground(Color.WHITE);
        whiteButton.addActionListener(this);
        buttonPanel2.add(whiteButton);

        JButton blueButton = new JButton("Blue");
        blueButton.setBackground(Color.BLUE);
        blueButton.addActionListener(this);
        buttonPanel2.add(blueButton);

        add(buttonPanel2, BorderLayout.SOUTH);
    }

    /**
     * Handles events generated by the buttons.
     *
     * @param e Specifies the generated event.
     */
    public void actionPerformed(ActionEvent e)
    {
        String buttonString = e.getActionCommand();

        if (buttonString.equals("Pink"))
            pinkPanel.setBackground(Color.PINK);
        else if (buttonString.equals("Green"))
            greenPanel.setBackground(Color.GREEN);
        else if (buttonString.equals("Yellow"))
            yellowPanel.setBackground(Color.YELLOW);
        else if (buttonString.equals("Red"))
            redPanel.setBackground(Color.RED);
        else if (buttonString.equals("White"))
            whitePanel.setBackground(Color.WHITE);
        else if (buttonString.equals("Blue"))
            bluePanel.setBackground(Color.BLUE);
        else
            System.out.println("Unexpected error.");
    }

```

Default window, and fully colored window.



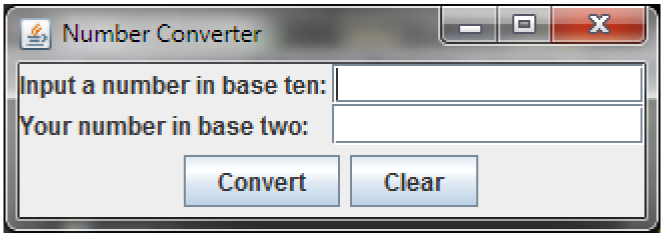
Demonstrating that each of the buttons affects only their own panels.



Name:	Dan Cassidy
Class:	CSCI-C 490, Mobile Application Development
Assignment:	Homework 3 Part 4
Date:	2015-07-13

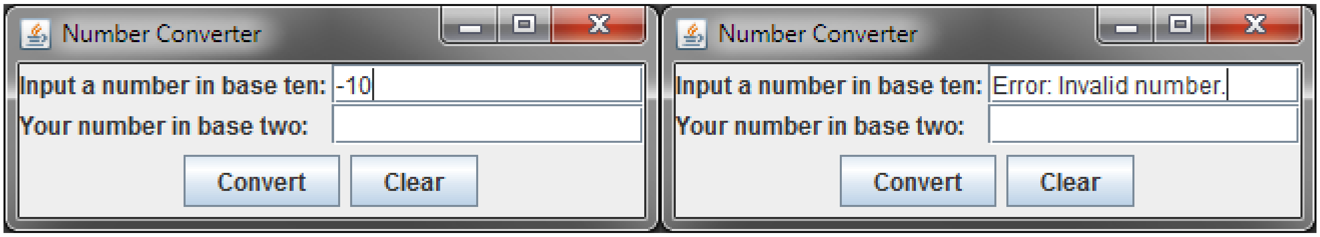
```java
/*--------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-07-13
 * Assignment:  HW3-4
 * Source File: NumberConverter.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
--------------------------------------------------------------------------------------------------*/
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 * Small GUI-based program to convert a base ten number into a base two number.
 *
 * @author Dan Cassidy
 */
public class NumberConverter extends JFrame implements ActionListener
{
    private JTextField textBaseTen;
    private JTextField textBaseTwo;

    /**
     * Entry point for the class.
     *
     * @param args Command line arguments. <i>Ignored</i>.
     */
    public static void main(String[] args)
    {
        NumberConverter gui = new NumberConverter();
        gui.setVisible(true);
    }

    /**
     * Default constructor. Handles the setup of all the GUI elements.
     */
    public NumberConverter()
    {
        super("Number Converter");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Set up the labels and input text boxes.
        JPanel inputPanel = new JPanel();
        inputPanel.setLayout(new GridLayout(2, 2));

        JLabel labelBaseTen = new JLabel("Input a number in base ten: ");
        inputPanel.add(labelBaseTen);

        textBaseTen = new JTextField();
        inputPanel.add(textBaseTen);

        JLabel labelBaseTwo = new JLabel("Your number in base two: ");
        inputPanel.add(labelBaseTwo);

        textBaseTwo = new JTextField();
        inputPanel.add(textBaseTwo);

        add(inputPanel, BorderLayout.NORTH);

        // Set up the buttons.
        JPanel buttonsPanel = new JPanel();
        buttonsPanel.setLayout(new FlowLayout());

        JButton buttonConvert = new JButton("Convert");
        buttonConvert.addActionListener(this);
        buttonsPanel.add(buttonConvert);

        JButton buttonClear = new JButton("Clear");
        buttonClear.addActionListener(this);
        buttonsPanel.add(buttonClear);

        add(buttonsPanel, BorderLayout.SOUTH);

        // Make the UI arrange itself.
        pack();
    }

    /**
     * Handles events generated by the buttons.
     *
     * @param e Specifies the generated event.
     */
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String buttonString = e.getActionCommand();

        // Convert button was clicked.
        if (buttonString.equals("Convert"))
        {
            try
            {
                textBaseTwo.setText(convertToBaseTwo(textBaseTen.getText()));
            }
            catch (IllegalArgumentException ex)
            {
                textBaseTen.setText("Error: Invalid number.");
            }
            catch (Exception ex)
            {
                textBaseTen.setText(ex.getMessage());
                ex.printStackTrace();
            }
            textBaseTen.requestFocus();
        }
        // Clear button was clicked.
        else if (buttonString.equals("Clear"))
        {
            textBaseTen.setText("");
            textBaseTwo.setText("");
            textBaseTen.requestFocus();
        }
        // Error.
        else
            System.out.println("Unexpected error.");
    }

    /**
     * Converts a base ten number into a base two number.
     *
     * @param inputNumber Contains the number to be converted from base ten to base two.
     * @return String, holding the number to be displayed.
     * @throws NumberFormatException if <b>inputNumber</b> cannot be parsed into an integer.
     * @throws IllegalArgumentException if <b>inputNumber</b> is negative.
     */
    private String convertToBaseTwo(String inputNumber)
    {
        String result = "";

        // Try to parse the input string, then check if the number is good. If parsing fails or the
        // number is bad, exceptions are thrown.
        int number = Integer.parseInt(inputNumber);
        if (number < 0)
            throw new IllegalArgumentException();

        // Handle the number.
        if (number == 0)
            result = "0";
        else
            while (number != 0)
            {
                result = (number % 2) + result;
                number /= 2;
            }

        return result;
    }

}
```

Default window.



Input a number in base ten:
Your number in base two:

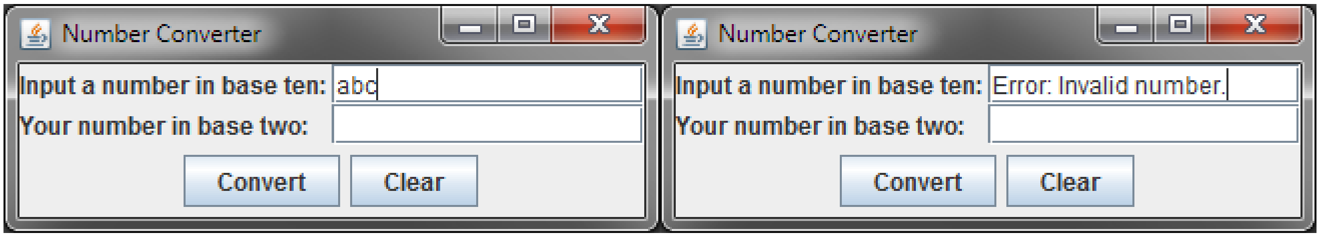
Demonstrating how the program handles negative values.



Input a number in base ten: -10
Your number in base two:

Input a number in base ten: Error: Invalid number.
Your number in base two:

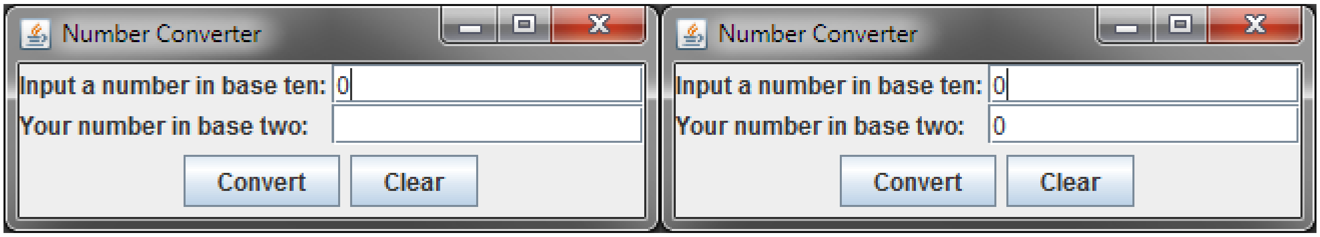
Demonstrating how the program handles bad input.



Input a number in base ten: abc
Your number in base two:

Input a number in base ten: Error: Invalid number.
Your number in base two:

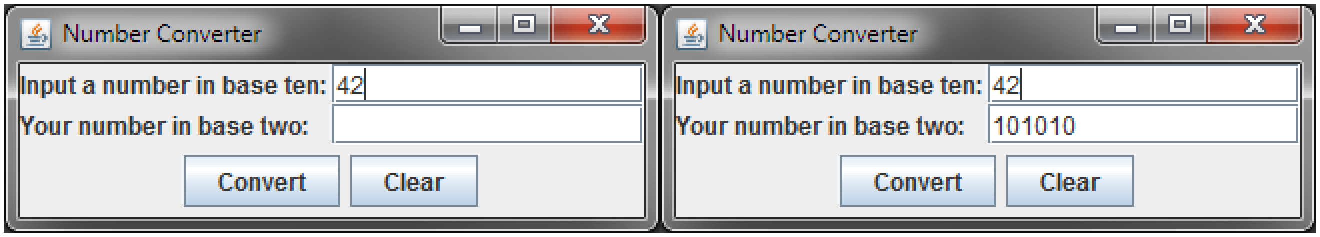
Demonstrating the edge case of ‘0’.



Input a number in base ten: 0
Your number in base two:

Input a number in base ten: 0
Your number in base two: 0

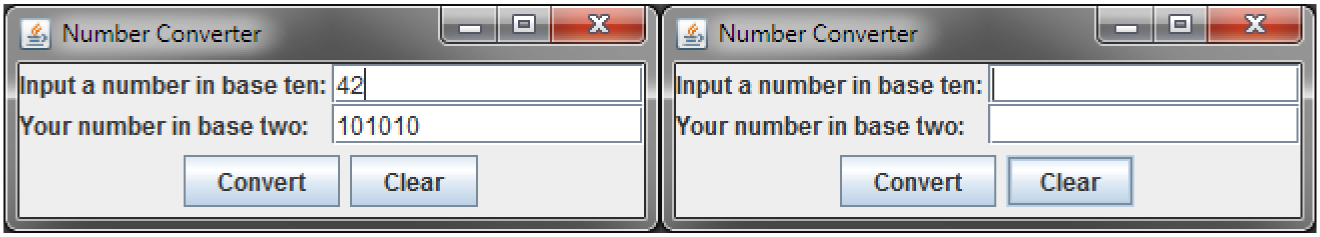
Demonstrating a normal case.



Input a number in base ten: 42
Your number in base two:

Input a number in base ten: 42
Your number in base two: 101010

Demonstrating the clear action.



Input a number in base ten: 42
Your number in base two: 101010

Input a number in base ten:
Your number in base two: