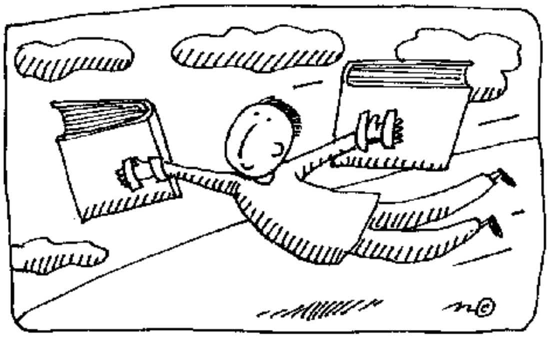
Level Up



DO NOT MAKE NEW SCREENS FOR EACH LEVEL.
Instead: reuse your screen for all of the levels!!

1. Add a new array called level1.

```
//globally
int level = 1; //a variable to track which level is displayed
int b [][] = original array
int level1 [][] = original array … exactly the same as b,
              but a different name.
```

   Level1's purpose is to store the original level 1 array.

2. Make other levels.

```
int level2 [][] = next level, same dimension as first.
int level3 [][] = next level, same dimension as first.
```

3. If you needed winning arrays, make new winning arrays for each level too. (See Check Answer).

```
int ans [][] = what the correct answer would be
int ans1 [][] = a copy of the original answer
int ans2 [][] = level 2's answer
int ans3 [][] = level 3's answer
```

4. This method copies any array 'n' into another array, thus resetting the array.

```
public void copyOver (int m[][], int n[][])
{ //copy every element of n into m
    //that changes the level
    for (int i = 0 ; i < row ; i++)
        for (int j = 0 ; j < col ; j++)
            m [i] [j] = n [i] [j];
}
```

5. Code levelUp. This calls copyOver to move the new arrays into b and ans.

```
public void levelUp ()
{
    level++;
    if (level == 1)
    {
        copyOver (b, level1);
        copyOver (ans, ans1);
    }
    else if (level == 2)
    {
        copyOver (b, level2);
        copyOver (ans, ans2);
    }
    else if (level == 3)
    {
        copyOver (b, level3);
        copyOver (ans, ans3);
    }
    else
    { //show your final win screen.
        cdLayout.show (p_card, "4");
    }

    //call redraw to update the screen
    redraw ();
}
```

6. Call `copyOver;` in your actionPerformed inside the proper else if clause. The if will already be present from CheckAnswer.

```
//inside the if clause, check for a win
if (check ())
{
    showStatus ("you win!");
    levelUp();
}
else
    showStatus ("keep going");
```