

EXERCISE SOLUTIONS

1.

```
int values[] = new int[100];
for (int i = 0; i < 100; i++)
    values[i] = i;
```
2.

```
String name[] = {"Alice","Bob","Charlie","Dave"};
```
3.

```
ArrayList<String> names = new ArrayList<String>();
names.add("Alice");
names.add("Bob");
names.add("Charlie");
names.add("Dave");
for (int i = 0; i < names.size(); i++)
    System.out.println(names.get(i));
names.set(1, "Betsy");
names.remove("Charlie");
for (String name : names)
    System.out.println(name);
```
4.

```
int[] values = { 3, 2, 7, 9, 10, 12, -5, 42, 101, 496};
int sum = 0;
for (int i = 0; i < values.length; i++)
    sum += values[i];
System.out.println(sum);
```
5.

```
ArrayList<Integer> values = new ArrayList<Integer>();
values.add(3); values.add(2); values.add(7); values.add(9);
values.add(10); values.add(12); values.add(-5); values.add(42);
values.add(101); values.add(496);
int count = 0;
// can use an indexed loop or enhanced for loop to count
for (Integer value : values)
{
    if (value % 2 == 0)
        count++;
}
System.out.println(count);
```
6.

```
int[][] table = new int[10][12];
for (int row = 0; row < table.length; row++)
    for (int col = 0; col < table[0].length; col++)
        table[row][col] = row * col;
```

7.

```
ArrayList<Integer> oneD = new ArrayList<Integer>();
for (int c = 0; c < 30; c++)
    oneD.add(c);
int[][] twoD = new int[5][6];
int i = 0; // used to count through the ArrayList
for (int row < twoD.length; row++)
{
    for (int col < twoD[0].length; col++)
    {
        twoD[row][col] = oneD.get(i);
        i++;
    }
}
```

8.

```
public static void playTicTacToe()
{
    Scanner in = new Scanner(System.in);
    String[][] board = new String[3][3];
    int plays = 0;
    while (plays < 9)
    {
        // Print board
        for (int row = 0; row < board.length; row++)
        {
            for (int col = 0; col < board[0].length; col++)
            {
                if (board[row][col] == null)
                    System.out.print(" . ");
                else
                    System.out.print(" " + board[row][col] + " ");
            }
            System.out.println();
        }

        System.out.print("Enter symbol (X or O): ");
        String symbol = in.next();
        System.out.print("Enter column: ");
        int col = in.nextInt();
        System.out.print("Enter row: ");
        int row = in.nextInt();

        if (row >= 0 && row < board.length &&
            col >= 0 && col < board[0].length &&
            board[row][col] == null)
        {
            board[row][col] = symbol;
            plays++;
        }
        else
            System.out.println("Invalid play. Try again.");
    }
}
```

9.

```
// Check to see if winner occurred in rows
for (int row = 0; row < board.length; row++)
{
    int playcount = 1;
    for (int col = 1; col < board[0].length; col++)
    {
        if (board[row][col].equals(board[row][0]))
            playcount++;
    }
    if (playcount == 3)
    {
        System.out.println(board[row][0] + " is the winner in rows!");
        break; // out of row loop
    }
}

// Check to see if winner occurred in columns
for (int col = 0; col < board[0].length; col++)
{
    int playcount = 1;
    for (int row = 1; row < board.length; row++)
    {
        if (board[row][col].equals(board[0][col]))
            playcount++;
    }
    if (playcount == 3)
    {
        System.out.println(board[0][col] + " is the winner in columns!");
        break; // out of col loop
    }
}

// Check downward-right diagonal
int playcount = 1;
for (int i = 1; i < board.length; i++)
{
    if (board[i][i].equals(board[0][0]))
        playcount++;
}
if (playcount == 3)
{
    System.out.println(board[0][0] + " is the winner on the downward-right
diagonal!");
}

// Check upward-right diagonal
playcount = 1;
for (int i = 1; i < board.length; i++)
{
    if (board[board.length - i - 1][i].equals(board[2][0]))
        playcount++;
}
if (playcount == 3)
{
    System.out.println(board[2][0] + " is the winner on the upward-right diagonal!");
}
}
```