1. Draw a picture showing the array contents after execution of the following code fragments.

```java
int[][] table = new int[2][4];
for (int i=0; i < table.length; i++)
    for (int j=0; j < table[i].length; j++)
        table[i][j] = i - j;
```

```
0 1 2 3
0 0 -1 -2 -3
1 1 0 -1 -2
```
2. Consider the following program:

```java
// ProductCodes.java       Author: Lewis/Loftus
import java.util.Scanner;
public class ProductCodes
{
    //--------------------------------------------------------------------------------
    //  Counts the number of product codes that are entered with a
    //  zone of R and and district greater than 2000.
    //--------------------------------------------------------------------------------
    public static void main(String[] args)
    {
        String code;
        char zone;
        int district, valid = 0, banned = 0;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter product code (XXX to quit): ");
        code = scan.nextLine();
        while (!code.equals("XXX"))
        {
            zone = code.charAt(9);
            district = Integer.parseInt(code.substring(3, 7));
            valid++;
            if (zone == 'R' && district > 2000)
                banned++;
            System.out.print("Enter product code (XXX to quit): ");
            code = scan.nextLine();
        }
        System.out.println("# of valid codes entered: " + valid);
        System.out.println("# of banned codes entered: " + banned);
    }
}
```

a) What kinds of errors in the input does the following catch clause handle? Give two examples of codes that would be caught and handled in this way

```
catch (StringIndexOutOfBoundsException exception)
{
    System.out.println("Bad code: " + code);
}
```

Catches errors arising from `code` not being long enough, thrown by `charAt()` and `substring()` methods. For example, if code is:

- TTTT
- AA

b) Annotate `ProductCodes.java` to show exactly how the try/catch block should be placed.
1. Draw a picture showing the array contents after execution of the following code fragments.

```java
int[][] table = new int[3][2];
for (int i=0; i < table.length; i++)
    for (int j=0; j < table[i].length; j++)
        table[i][j] = i + j;
```

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
```
2. Consider the following program:

```java
// ProductCodes.java       Author: Lewis/Loftus
import java.util.Scanner;
public class ProductCodes
{
    // Counts the number of product codes that are entered with a
    // zone of R and and district greater than 2000.

    public static void main(String[] args)
    {
        String code;
        char zone;
        int district, valid = 0, banned = 0;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter product code (XXX to quit): ");
        code = scan.nextLine();
        while (!code.equals("XXX"))
        {
            zone = code.charAt(9);
            district = Integer.parseInt(code.substring(3, 7));
            valid++;
            if (zone == 'R' && district > 2000)
                banned++;
            System.out.print("Enter product code (XXX to quit): ");
            code = scan.nextLine();
        }
        System.out.println("# of valid codes entered: "+ valid);
        System.out.println("# of banned codes entered: " + banned);
    }
}
```

a) What kinds of errors in the input does the following catch clause handle? Give two examples of codes that would be caught and handled in this way

```java
catch (NumberFormatException exception)
{
    System.out.println("Bad code: " + code);
}
```

Catches errors arising from `code` not containing an integer in positions 3-6, thrown by `parseInt()`. For example, if code is:

* TTTTTTTTTTTTTTT
* AAA34X6AAAAAAAAAAA

b) Annotate `ProductCodes.java` to show exactly how the try/catch block should be placed.