Case Study: Lotto Numbers

<key point>

The problem is to write a program that checks if all the input numbers cover `1` to `99`.

<end key point>

Each ticket for the Pick-10 lotto has `10` unique numbers ranging from `1` to `99`. Suppose you buy a lot of tickets and like to have them cover all numbers from `1` to `99`. Write a program that reads the ticket numbers from a file and checks whether all numbers are covered. Assume the last number in the file is `0`. Suppose the file contains the numbers

```
80 3 87 62 30 90 10 21 46 27
12 40 83 9 39 88 95 59 20 37
80 40 87 67 31 90 11 24 56 77
11 48 51 42 8 74 1 41 36 53
52 82 16 72 19 70 44 56 29 33
54 64 99 14 23 22 94 79 55 2
60 86 34 4 31 63 84 89 7 78
43 93 97 45 25 38 28 26 85 49
47 65 57 67 73 69 32 71 24 66
92 98 96 77 6 75 17 61 58 13
35 81 18 15 5 68 91 50 76
0
```

Your program should display

```
The tickets cover all numbers
```

Suppose the file contains the numbers

```
11 48 51 42 8 74 1 41 36 53
52 82 16 72 19 70 44 56 29 33
0
```

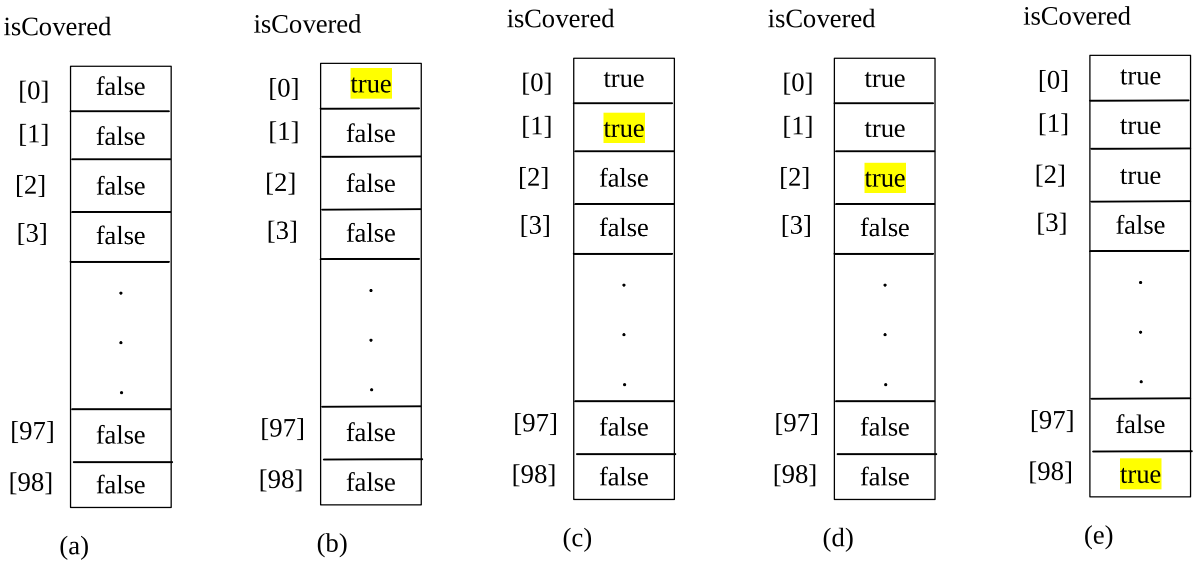
Your program should display

```
The tickets don't cover all numbers
```

How do you mark a number as covered? You can create an array with `99` `boolean` elements. Each element in the array can be used to mark whether a number is covered. Let the array be `isCovered`. Initially, each element is false, as shown in Figure 6.2a. Whenever a number is read, its corresponding element is set to true. Suppose the numbers entered are 1, 2, 3, 99, 0. When number 1 is read, isCovered[0] is set to true (see Figure 6.2b). When number 2 is read, isCovered[2 - 1] is set to true (see Figure 6.2c). When number 3 is read, isCovered[3 - 1] is set to true (see Figure 6.2d). When number 99 is read, isCovered[98] is set to true (see Figure 6.2e).

Figure 6.2

If number i *appears in a Lotto ticket,* isCovered[i-1] *is set to* true.



The algorithm for the program can be described as follows:

```
for each number k read from the file,
  mark number k as covered by setting isCovered[k – 1] true;
if every isCovered[i] is true
  The tickets cover all numbers
else
  The tickets don’t cover all numbers
```

The complete program is given in Listing 6.1.

Listing 6.1 LottoNumbers.java

```
 1  import java.util.Scanner;
 2
 3  public class LottoNumbers {
 4    public static void main(String[] args) {
 5      Scanner input = new Scanner(System.in);
 6      boolean[] isCovered = new boolean[99]; // Default is false
 7
 8      // Read each number and mark its corresponding element covered
 9      int number = input.nextInt();
10      while (number != 0) {
11        isCovered[number - 1] = true;
12        number = input.nextInt();
13      }
14
15      // Check whether all covered
16      boolean allCovered = true; // Assume all covered initially
17      for (int i = 0; i < isCovered.length; i++)
18        if (!isCovered[i]) {
19          allCovered = false; // Find one number not covered
20          break;
21        }
22
23      // Display result
24      if (allCovered)
25        System.out.println("The tickets cover all numbers");
26      else
27        System.out.println("The tickets don’t cover all numbers");
28    }
29  }
```

*<margin note (line 6)>*create and initialize array

*<margin note (line 9)>*read number

*<margin note (line 11)>*mark number covered

<margin note (line 12)>read number

<margin note (line 24)>check allCovered?

<end listing 6.1>

Suppose you have created a text file named LottoNumbers.txt that contains the input data `2 5 6 5 4 3 23 43 2 0`. You can run the program using the following command:

```
    java LottoNumbers < LottoNumbers.txt
```

The program can be traced as follows:

<trace program>

Line#	**Representative elements in array `isCovered`**							**`number`**	**`allCovered`**
	[1]	**[2]**	**[3]**	**[4]**	**[5]**	**[22]**	**[42]**		
6	false	false	false	false	false	false	false		
9								2	
11	true								
12								5	
11				true					
12								6	
11					true				
12								5	
11				true					
12								4	
11			true						
12								3	
11		true							
12								23	
11						true			
12								43	
11							true		
12								2	
11	true								

12								0	
16									true
18(i=0)									false

<end trace program>

The program creates an array of `99` `boolean` elements and initializes each element to `false` (line 6). It reads the first number from the file (line 9). The program then repeats the following operations in a loop:

- If the number is not zero, set its corresponding value in array `isCovered` to `true` (line 11);
- Read the next number (line 12).

When the input is `0`, the input ends. The program checks whether all numbers are covered in lines 16–21 and displays the result in lines 24–27.