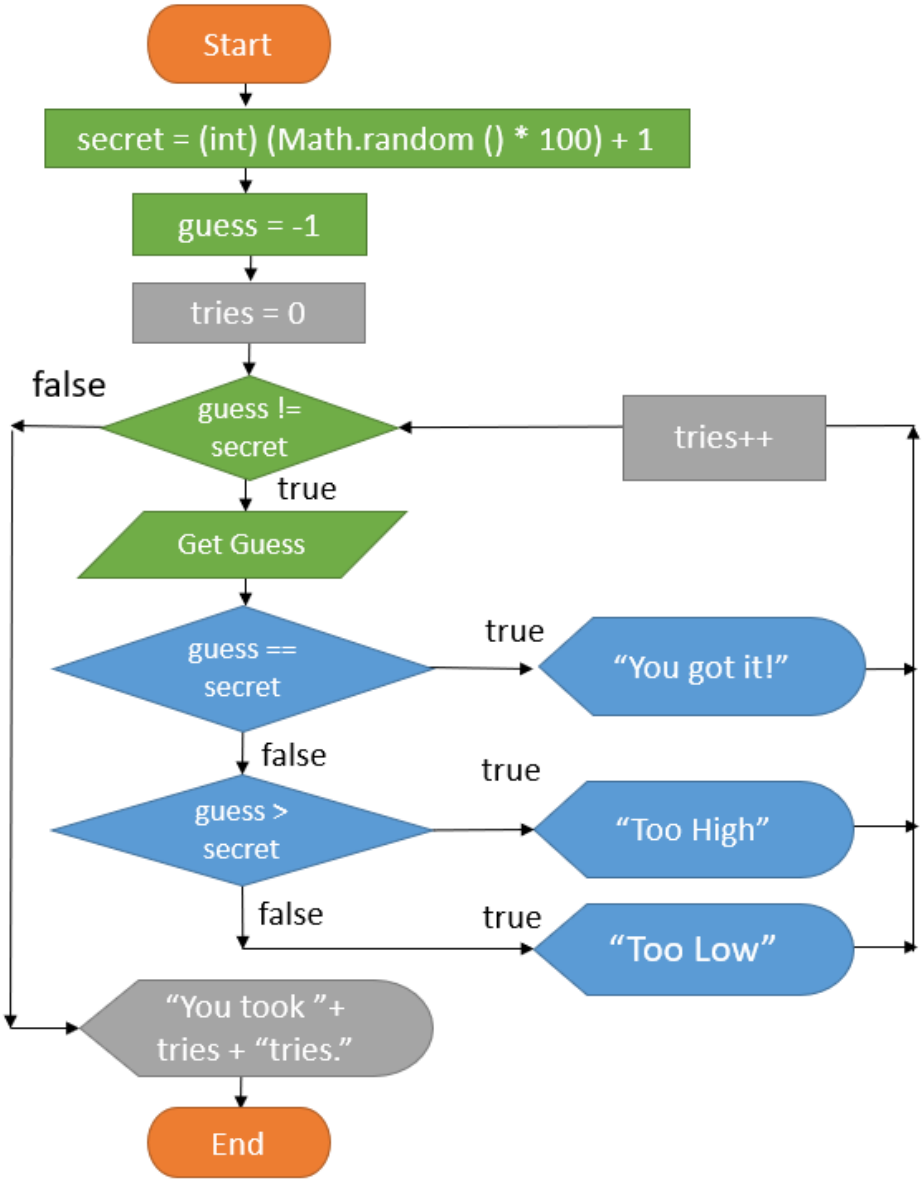
Guessing Game

In the Guessing Game, the computer picks a number between 1 and 100. It gives you clues until you are able to guess the exact number.

Take this flowchart and make it into a program.



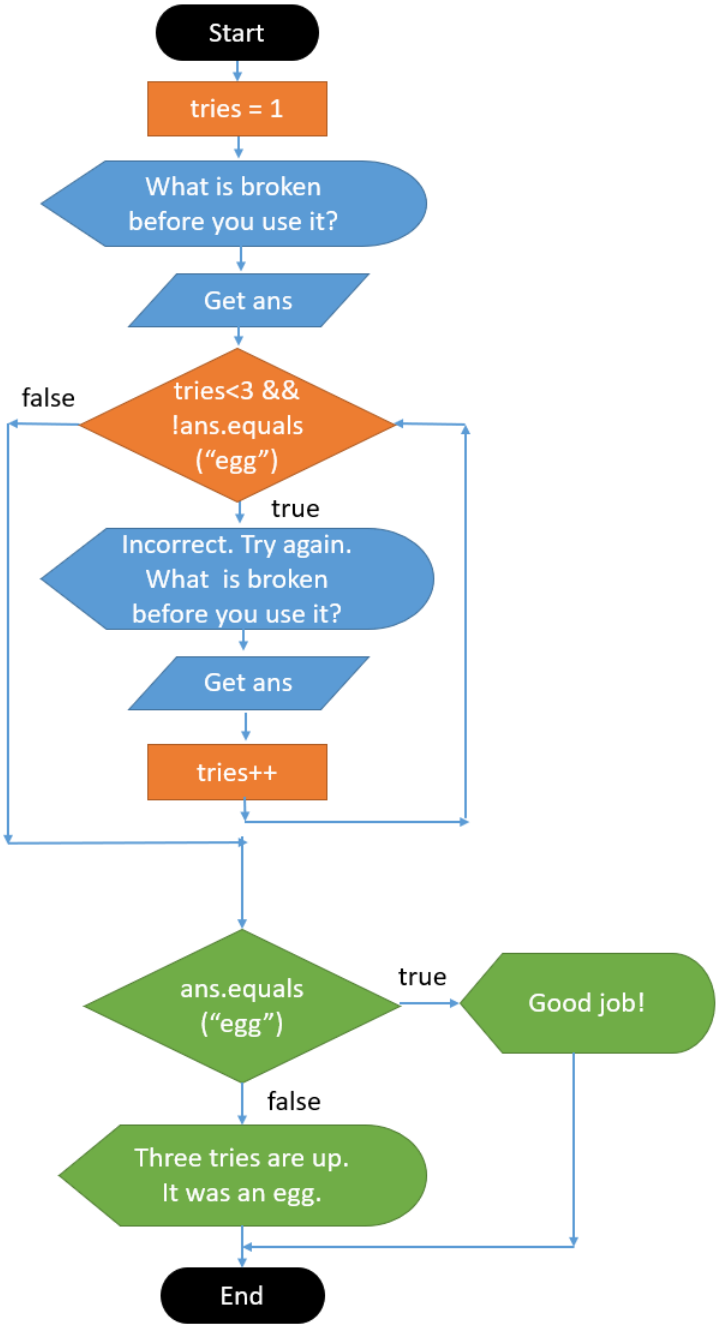
Game Play looks like this:

```
Guess? 50
Too Low!
Guess? 75
Too Low!
Guess? 87
Too High!
Guess? 82
Too Low!
Guess? 84
Too Low!
Guess? 86
Too High!
Guess? 85
You got it!
You took 7 tries.
```

This is the code needed to make a random number.

```
int secret = (int) (Math.random () * 100) + 1;
```

Riddle Program



This program asks the user a riddle and gives them three chances to guess it.

If they don't get it in three chances, the program tells the user the answer.

Create a program using the flowchart.

One run of the program:

```
What is broken before you use it? staple
Incorrect. Try again
What is broken before you use it? pencil
Incorrect. Try again
What is broken before you use it? eraser
Three tries are up. It was an egg.
```

Another run of the program:

```
What is broken before you use it? banana
Incorrect. Try again
What is broken before you use it? egg
Good job!
```

I Speak TXTMSG

Text messaging using a cell phone is popular among teenagers (you are shocked, I know). The messages can appear peculiar because short forms and symbols are used to abbreviate messages and hence reduce typing. For example, “LOL” means “laughing out loud” and “:-)” is called an emoticon which looks like a happy face (on its side) and it indicates chuckling. This is all quite a mystery to some adults.

Write a program that will continually input a short form and output the translation for an adult using the following translation table:

Short Form	Translation
CU	see you
:-)	I’m happy
:-(	I’m unhappy
;-)	wink
:-P	stick out my tongue
(˜.˜)	sleepy
TA	totally awesome
CUZ	because
TY	thank-you
YW	you're welcome
TTYL	talk to you later

Input Specifications

The user will be prompted to enter text to be translated one line at a time. When the short form “TTYL” is entered, the program ends. Users may enter text that is found in the translation table, or they may enter other words. All entered text will be symbols or upper case letters. There will be no spaces and no quotation marks.

Output Specifications

The program will output text immediately after each line of input. If the input is one of the phrases in the translation table, the output will be the translation; if the input does not appear in the table, the output will be the original word. The translation of the last short form entered “TTYL” should be output.

Sample Session (user input is in italics)

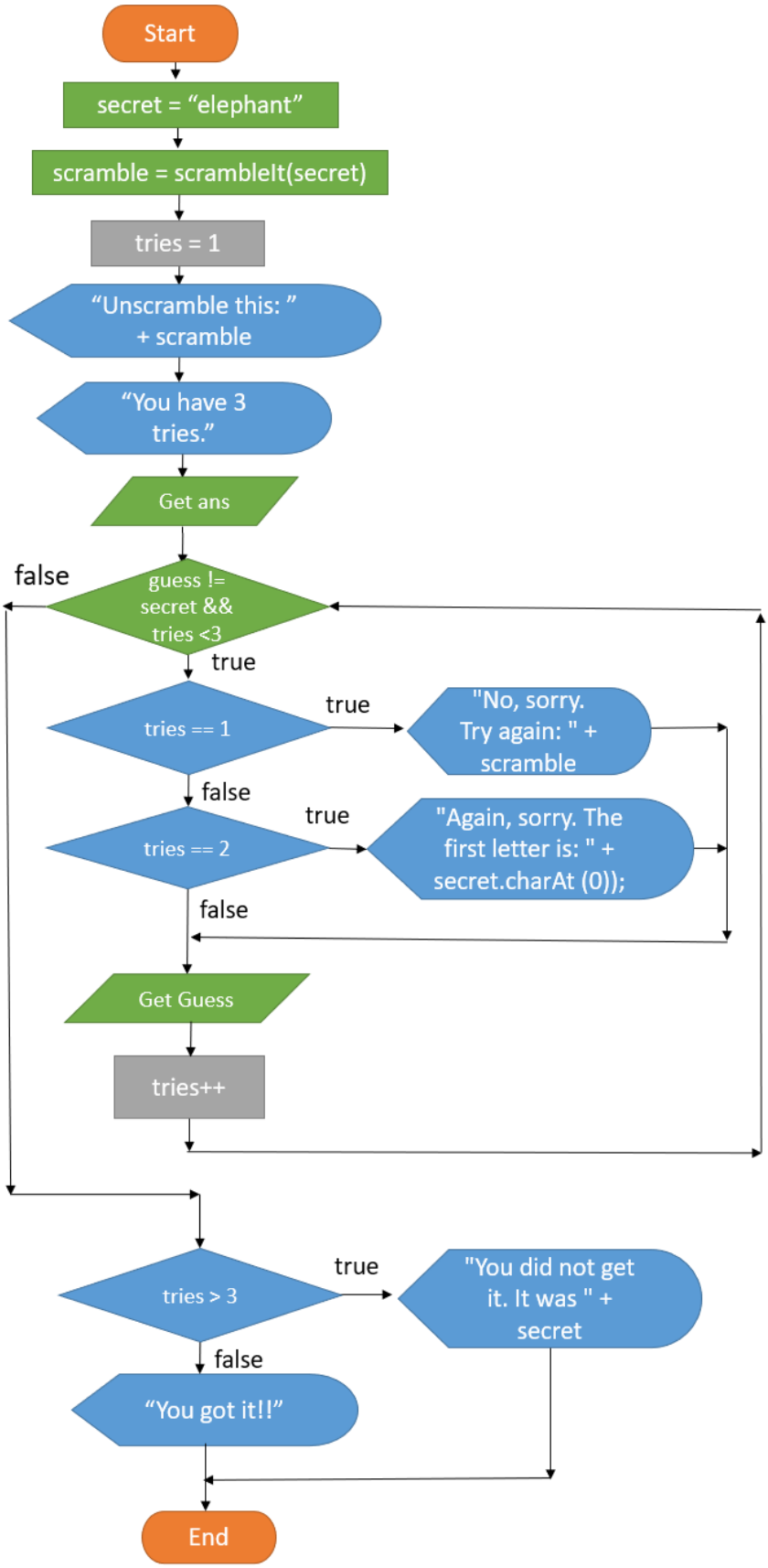
```
Enter phrase> YW
you're welcome
Enter phrase> :-)
I’m happy
Enter phrase> SQL
SQL
Enter phrase> TTYL
talk to you later
```

from http://cemc.uwaterloo.ca/contests/computing/2007/index.html, Junior Problems

Unscramble

The flowchart for Unscramble is on the right.

Use it to fill in the blanks in the starter code on the next page.



Unscramble Starter Code

```
public class UnscrambleTask
{
    public static void main (String args[])
    {
        new UnscrambleTask ();
    }

    public UnscrambleTask ()
    { //set up answer (secret) and question (scramble)
        String secret = "__________";
        String scramble = scrambleIt (secret);
        int tries = _____;

        //ask the question, get an answer
        System.out.println (_____________);
        System.out.println (_____________);
        String ans = IO.inputString ("Answer? ");

        //handle the wrong values FIRST
        while (!secret.equalsIgnoreCase (ans) && tries < 3)
        { //give clues based on number of tries
            if (_______)
                System.out.println (____________);
            else if (_________)
                System.out.println ("Again, sorry. The first letter is: " + secret.charAt (0));
            //get the next guess
            ans = IO.inputString ("Answer? ");
            _______________
        }

        //After the loop, determine win or not
        if (_____________)
        {
            System.out.println ("You did not get it. It was " + secret);
        }
        else
        {
            System.out.println ("_________________");
        }
    }


    public String scrambleIt (String secret)
    { //Method used to scramble a word

        String scramble = secret;
        //Randomly slice up the word 20 times
        for (int i = 0 ; i < 20 ; i++)
        {
            //pick 3 spots in the word
            int c = secret.length ();
            int a = (int) (Math.random () * c);
            int b = (int) (Math.random () * c);
            //a needs to be less than b, guarantee it
            if (a > b)
            {
                int temp = b;
                b = a;
                a = temp;
            }
            //slice into 3
            String first = scramble.substring (0, a);
            String mid = scramble.substring (a, b);
            String last = scramble.substring (b, c);
            //scramble the 3 pieces
            scramble = last + mid + first;
        }
        return scramble;
    }
}
```