1- Right Triangle Checker

Write a program to allow the user to enter three integers. These integers need to be in ascending (increasing) order, though duplicate numbers are allowed.

The first two numbers are the measure of the sides and the third is the hypotenuse. Print whether the three sides make a right triangle or not.

```
Enter three integers in ascending order:
Side 1: 4
Side 2: 3
3 is smaller than 4. Try again.
Side 2: -9
-9 is smaller than 4. Try again.
Side 2: 5
Side 3: 1
1 is smaller than 5. Try again.
Side 3: 5

Your three sides are 4 5 5
No! These sides do not make a right triangle!

Enter three integers in ascending order:
Side 1: 6
Side 2: 8
Side 3: 10
Your three sides are 6 8 10
These sides *do* make a right triangle.
```

2- Collatz Sequence

Take any natural number n.

If n is even, divide it by 2 to get n / 2.
If n is odd, multiply it by 3 and add 1 to get 3n + 1.
Repeat the process indefinitely.

In 1937, Lothar Collatz proposed that no matter what number you begin with, the sequence eventually reaches 1. This is widely believed to be true, but has never been formally proved.

Write a program that inputs a number from the user, and then displays the Collatz Sequence starting from that number. Stop when you reach 1.
Also display the total number of steps in the sequence.
Also display the largest value encountered in the sequence.

```
Enter starting number: 11
11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Terminated after 14 steps. The largest value was 52
```

```
Enter starting number: 27
27 82 41 ............... 1
Terminated after 111 steps. The largest value was 9232
```

3- Multiplication Table

Use nested for loops to generate a multiplication table,
which should go all the way up to 10x10.Output should look like
the table shown below.

```
 x |  1  2  3  4  5  6  7  8  9 10
===+=================================
 1 |  1  2  3  4  5  6  7  8  9 10
 2 |  2  4  6  8 10 12 14 16 18 20
 ..
 9 |  9 18 27 36 45 54 63 72 81 90
10 | 10 20 30 40 50 60 70 80 90 100
```