**Write C# code to declare** a variable to store the age of a person. Then the output of the program is as an example shown below:

```
You are 20 years old.
```

---

**Write a C# program that** prompts the user to input three integer values and find the greatest value of the three values

The program will read three types of scores(quiz score, mid-term score, and final score) and determine the grade based on the following rules:
-if the average score =90%                        : then display grade-A
-if the average score >= 70% and <90%   :then display grade-B
-if the average score>=50% and <70%     :then display grade-C
-if the average score<50%                        :then display grade-F

---

**Write C# code to prompt a user to enter an integer value.**

The value is stored in a variable called a. Then the program will show the output as seen below:

```
The value of a is 10.
...............................
The value of ++a is 11.
The value of a is 11.
The value of a++ is 11.
The value of a is 12.
The value of --a is 11.
The value of a is 11.
The value of a-- is 11.
The value of a is 10.
```

---

# Celsius to Fahrenheit

In this exercise, you will write a program to convert temperatures from Celsius to Fahrenheit.

The first step to solving this problem is to figure out how you can do this conversion by hand, with pen and pencil. Once you learn how to do it by hand, you can write a program that prompts the user for a temperature to convert.

The program can be broken into these steps:

- **First**, display a prompt to the user to have them enter *Degrees in Celsius*
- **Second**, listen for user input and store the value in a variable
- **Third**, convert the input into a number and apply a mathematical formula to it
- **Fourth**, display the result of the mathematical formula

---

**Write a C# program to detect key presses.**

If the user pressed number keys( from 0 to 9), the program will display the number that is pressed, otherwise the program will show "Not allowed"

---

**Write C# code to produce the output shown below**

```
*******
******
*****
****
***
**
*
```

---

Calculating the Ordinal

In this exercise, you will write a program to determine the ordinal of a number and display it on the screen.

The ordinal of a number is the number that defines a thing's position in a series, such as "first," "second," or "third." Our program will convert numbers like 1, 2, 3 into ordinals like 1st, 2nd, 3rd, etc.

Your program will accept user input and display information back to the user. This problem is actually a problem you may encounter when building real web applications.

So for example, given an input of 51, a message like *"That's the 51st item!"* should be displayed on the screen.

**Here are some hints to get you started:**

- **First**, prompt the user with the message, *"Enter a number"*
- **Second**, receive the user input and store it in a variable. Since you're dealing with numbers, ***you may want to convert it to an integer*** at this step.

- **Third**, lop off the right-most digit and store it in a variable.
    - **HINT!** A trick you can use to lop off the right-most digit is to calculate the number's % 10.
        - **For example:** 51 % 10 = 1
        - **And another example:** 75 % 10 = 5
- **Fourth**, depending on the right-most digit (equal to the input number modulo 10) that comes back, determine the suffix and display a message to the user.
    - If the right-most digit equals 1, the suffix should be st
    - Otherwise, if the digit equals 2, the suffix should be nd
    - Otherwise, if the digit equals 3, the suffix should be rd
    - Otherwise, the suffix should be th

---

# Foobar Problem

In this challenge exercise, you will solve a problem that is actually commonly asked during technical interviews and in the technical screening process.

This question is a ***REAL*** question that is actually asked in interviews. Being able to answer the Foobar problem will be important if you're looking to land a job, because it requires that you have a foundation with programming and the language you're using.

This problem has its origins in a children's game for learning about division, remainders and fractions. Teachers have schoolchildren sit in a circle, and the person who starts the game says the number "1." Each student in the circle has a turn to say the next number in a pattern. The catch is: for all numbers that are divisible by three, the player should say "foo." For any number divisible by five, the player should say "bar." Numbers divisible by both become "Foobar."

For example, a typical round of Foobar would start as follows:

1, 2, Foo, 4, Bar, Foo, 7, 8, Foo, Bar, 11, Foo, 13, 14, Foobar, 16, 17, Foo, 19, Bar, Foo, 22, 23, Foo, Bar, 26, Foo, 28, 29, Foobar, 31, 32, Foo, 34, Bar, Foo, ...

**In this problem, you should write a program to play the Foobar game and display n elements of the Foobar pattern!**

**Hints**

- **First**, prompt the user for how many digits of the pattern do they want to see.
- **Second**, since they entered a number, we'll need to convert it to an integer.
- **Third**, add the Foobar logic that will generate that many digits of the pattern.

Here's an example execution of the program. When we run our program, as a user we will see:

```
How many items do you want to see?
```

Then, as a user I type the number 7 and press enter:

```
How many items do you want to see? 7
```

As a user I will be presented the first 7 items in the Foobar pattern:

```
1
 2
 Foo
 4
 Bar
 Foo
 7
```

But if we started the program expecting only 4 elements like this:

```
How many items do you want to see? 4
```

We would be presented with this output after our program runs:

```
1
 2
 Foo
 4
```

---