

## Ejercicio 8

Luego de ejecutar el siguiente código: cuánto vale las variables cont y n antes de entrar al bucle y después de salir del bucle? En este ejercicio debe utilizar el depurador y la ventana watch para capturar (como imagen) el valor de las variables cont y n. Responda: ¿De forma general que cree que hace este programa? Haga pruebas ingresando los valores 5, 10, 123, 678, 899, 1000, 7890, 12789.

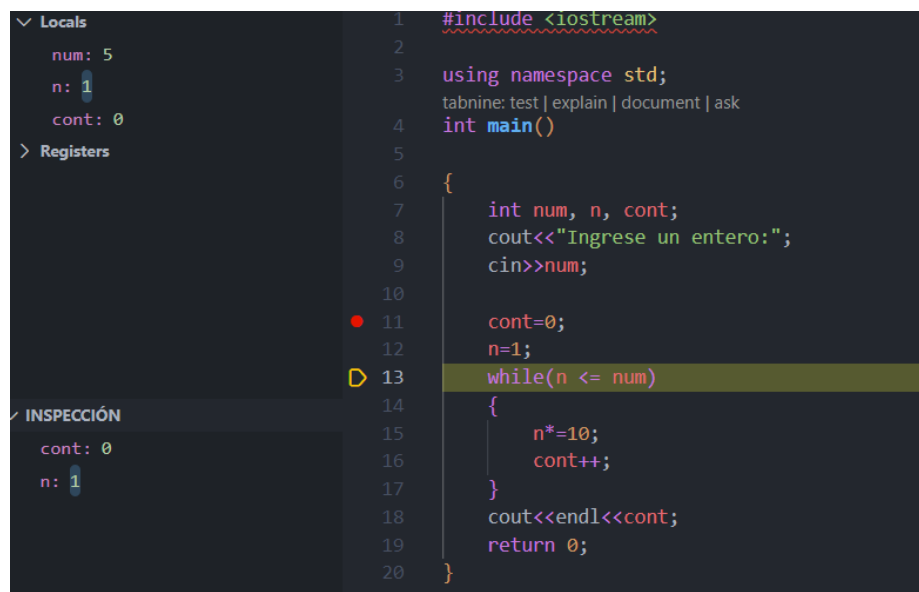
```
5  ∨ int main()
6  {
7      int num, n, cont;
8      cout<<"Ingrese un entero: ";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     ∨ while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

**¿De forma general que cree que hace este programa?**

El programa solicita al usuario ingresar un número entero, luego a través del bucle while multiplica la variable “n” por 10 hasta que esta misma variable sea mayor que el número anteriormente ingresado, mientras se incrementa el contador en cada iteración del bucle e imprime el valor de “cont” que representa al número de cifras del número entero ingresado.

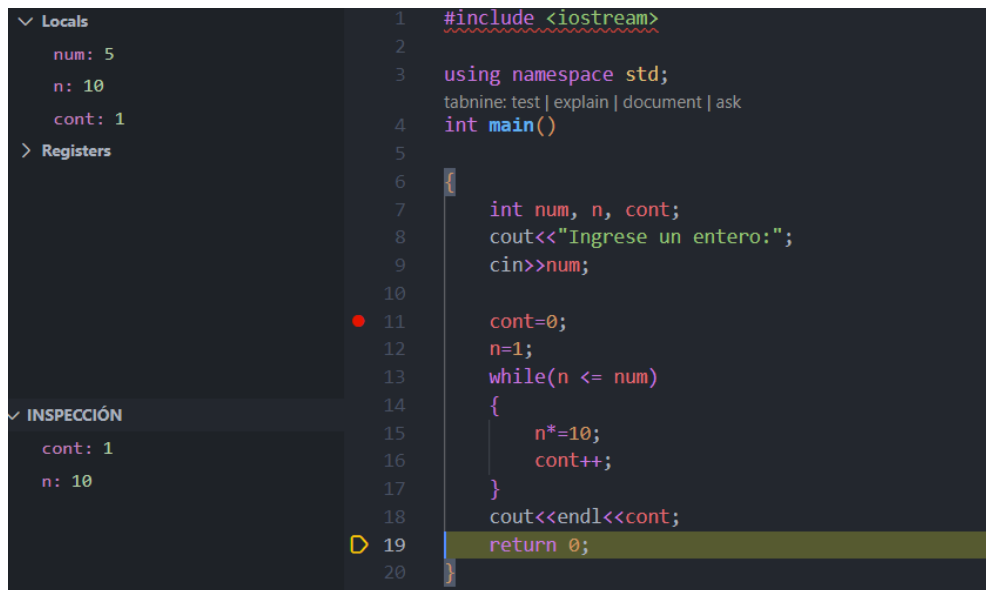
**Antes de entrar al bucle cuando el entero es 5:**

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1



### Después de salir del bucle cuando el entero es 5:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 5, tienen los siguientes valores cont = 1 y la variable n = 10



The screenshot shows a C++ IDE with a code editor and a variable inspector. The code is as follows:

```
1 #include <iostream>
2
3 using namespace std;
4 int main()
5 {
6
7     int num, n, cont;
8     cout<<"Ingrese un entero:";
9     cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

The variable inspector on the left shows the following values:

- Locals: num: 5, n: 10, cont: 1
- Registers: (empty)
- INSPECCIÓN: cont: 1, n: 10

The code editor shows the execution state at line 19, where the while loop has finished. A red dot is on line 11, and a yellow arrow points to line 19.

### Antes de entrar al bucle cuando el entero es 10:

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1



The screenshot shows the same C++ IDE as before, but with the execution state at line 13, just before the while loop. The code is the same as in the previous screenshot.

The variable inspector on the left shows the following values:

- Locals: num: 10, n: 1, cont: 0
- Registers: (empty)
- INSPECCIÓN: cont: 0, n: 1

The code editor shows the execution state at line 13, where the while loop is about to start. A red dot is on line 11, and a yellow arrow points to line 13.

### Después de salir del bucle cuando el entero es 10:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 10, tienen los siguientes valores cont = 2 y la variable n = 100

The screenshot shows a C++ IDE with a dark theme. On the left, the 'Locals' panel displays the current values of variables: num: 10, n: 100, and cont: 2. Below it, the 'Registers' panel is empty. The 'INSPECCIÓN' panel shows the same values: cont: 2 and n: 100. The main code editor on the right shows the following code:

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

The cursor is positioned at line 20, indicating the end of the program execution.

### Antes de entrar al bucle cuando el entero es 123:

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1

The screenshot shows the same C++ IDE as before, but with the cursor positioned at line 13, just before the while loop. The 'Locals' panel on the left shows the current values: num: 123, n: 1, and cont: 0. The 'Registers' panel is empty. The 'INSPECCIÓN' panel shows the same values: cont: 0 and n: 1. The main code editor on the right shows the same code as before, but the cursor is at line 13:

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

### Después de salir del bucle cuando el entero es 123:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 123, tienen los siguientes valores cont = 3 y la variable n = 1000

The screenshot shows a C++ IDE with a dark theme. On the left, the 'Locals' panel displays the current values of variables: num: 123, n: 1000, and cont: 3. Below it, the 'Registers' panel is empty. The 'INSPECCIÓN' panel shows the same values: cont: 3 and n: 1000. The main code editor shows the following code:

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

The code is at line 13, inside the while loop. A red dot is on line 11, and a yellow arrow points to line 20.

### Antes de entrar al bucle cuando el entero es 678:

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1

The screenshot shows the same C++ IDE as before, but with the input num set to 678. The 'Locals' panel shows num: 678, n: 1, and cont: 0. The 'Registers' panel is empty. The 'INSPECCIÓN' panel shows cont: 0 and n: 1. The main code editor shows the same code as before, but the while loop condition is now n <= 678. The code is at line 13, inside the while loop. A red dot is on line 11, and a yellow arrow points to line 13.

### Después de salir del bucle cuando el entero es 678:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 678, tienen los siguientes valores cont = 3 y la variable n = 1000

The screenshot shows a C++ IDE with a dark theme. On the left, the 'Locals' panel displays the current values of variables: num: 678, n: 1000, and cont: 3. Below it, the 'Registers' panel is empty. The 'INSPECCIÓN' panel also shows cont: 3 and n: 1000. The main editor shows the source code of 'Ejercicio\_8.cpp'. The code includes <iostream>, uses namespace std, and defines a main function. Inside main, it declares int num, n, cont; prompts the user for an integer and reads it into num. It then initializes cont=0 and n=1, and enters a while loop with condition (n <= num). Inside the loop, it multiplies n by 10 and increments cont. The loop ends at line 13, and the program prints the final value of cont (3) and returns 0.

```
1 #include <iostream>
2
3 using namespace std;
4 int main()
5 {
6
7     int num, n, cont;
8     cout<<"Ingrese un entero:";
9     cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

### Antes de entrar al bucle cuando el entero es 899:

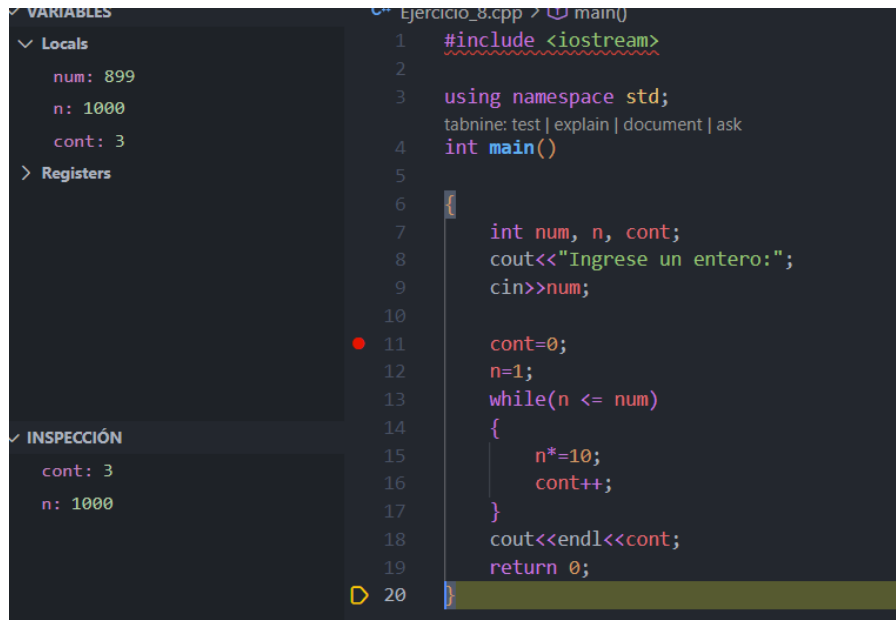
Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1

The screenshot shows the same C++ IDE as before, but with the program paused at line 13, just before the while loop. The 'Locals' panel shows num: 899, n: 1, and cont: 0. The 'INSPECCIÓN' panel shows cont: 0 and n: 1. The main editor shows the source code with line 13 highlighted. The code is identical to the previous screenshot, but the execution has stopped at the start of the while loop.

```
1 #include <iostream>
2
3 using namespace std;
4 int main()
5 {
6
7     int num, n, cont;
8     cout<<"Ingrese un entero:";
9     cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

### Después de salir del bucle cuando el entero es 899:

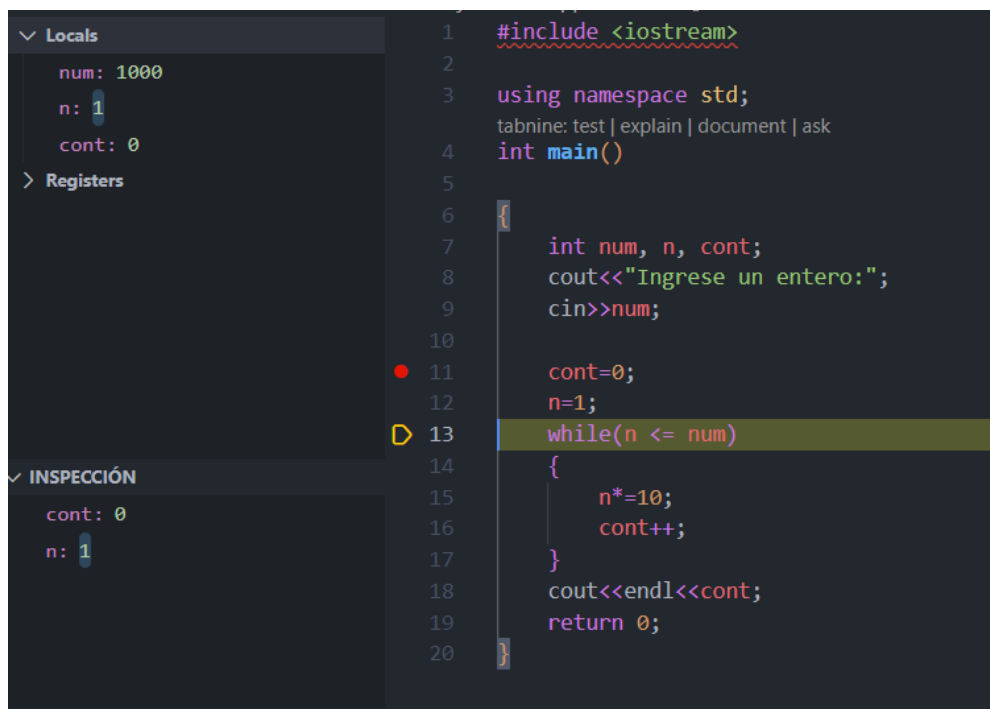
Las variables “cont” y “n” después de salir del bucle cuando el entero es 899, tienen los siguientes valores cont = 3 y la variable n = 1000



```
1 #include <iostream>
2
3 using namespace std;
4 int main()
5 {
6     int num, n, cont;
7     cout<<"Ingrese un entero:";
8     cin>>num;
9
10    cont=0;
11    n=1;
12    while(n <= num)
13    {
14        n*=10;
15        cont++;
16    }
17    cout<<endl<<cont;
18    return 0;
19 }
```

### Antes de entrar al bucle cuando el entero es 1000:

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1



```
1 #include <iostream>
2
3 using namespace std;
4 int main()
5 {
6     int num, n, cont;
7     cout<<"Ingrese un entero:";
8     cin>>num;
9
10    cont=0;
11    n=1;
12    while(n <= num)
13    {
14        n*=10;
15        cont++;
16    }
17    cout<<endl<<cont;
18    return 0;
19 }
```

### Después de salir del bucle cuando el entero es 1000:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 1000, tienen los siguientes valores cont = 4 y la variable n = 10000

The screenshot shows a C++ IDE with a dark theme. On the left, the 'Locals' panel displays the current values of variables: num: 1000, n: 10000, and cont: 4. Below it, the 'Registers' panel is empty. The 'INSPECCIÓN' (Inspection) panel shows the values of 'cont' as 4 and 'n' as 10000. The main code editor on the right shows a C++ program. The code includes `<iostream>`, uses the `std` namespace, and defines a `main` function. Inside `main`, it declares `int num, n, cont;`, prompts the user to enter an integer, and reads it into `num`. It then initializes `cont=0;` and `n=1;` and enters a `while(n <= num)` loop. Inside the loop, it multiplies `n` by 10 and increments `cont`. The loop terminates at line 13, and the final state of the variables is reflected in the left-hand panels.

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

### Antes de entrar al bucle cuando el entero es 7890:

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1

The screenshot shows the same C++ IDE as before, but at a different point in the program's execution. The 'Locals' panel now shows num: 7980, n: 1, and cont: 0. The 'INSPECCIÓN' panel shows cont: 0 and n: 1. In the code editor, the `while(n <= num)` loop at line 13 is highlighted, indicating it is about to be entered. The rest of the code remains the same as in the previous screenshot.

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

### Después de salir del bucle cuando el entero es 7890:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 7980, tienen los siguientes valores cont = 4 y la variable n = 10000

The screenshot shows a C++ IDE with a dark theme. On the left, the 'Locals' panel displays the current values of variables: num: 7980, n: 10000, and cont: 4. Below it, the 'Registers' panel is empty. The 'INSPECCIÓN' panel shows the same values: cont: 4 and n: 10000. The main code editor on the right shows a C++ program. The code includes `<iostream>`, uses the `std` namespace, and defines a `main` function. Inside `main`, it declares `int num, n, cont;`, prompts the user to enter an integer, and reads it into `num`. It then initializes `cont=0;` and `n=1;` and enters a `while(n <= num)` loop. Inside the loop, it multiplies `n` by 10 and increments `cont`. After the loop, it prints `endl<<cont;` and returns 0. A red dot on line 11 indicates the current execution point, which is after the loop has finished.

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

### Antes de entrar al bucle cuando el entero es 12789:

Las variables “cont” y “n” antes de entrar al bucle tienen los siguientes valores cont = 0 y la variable n = 1

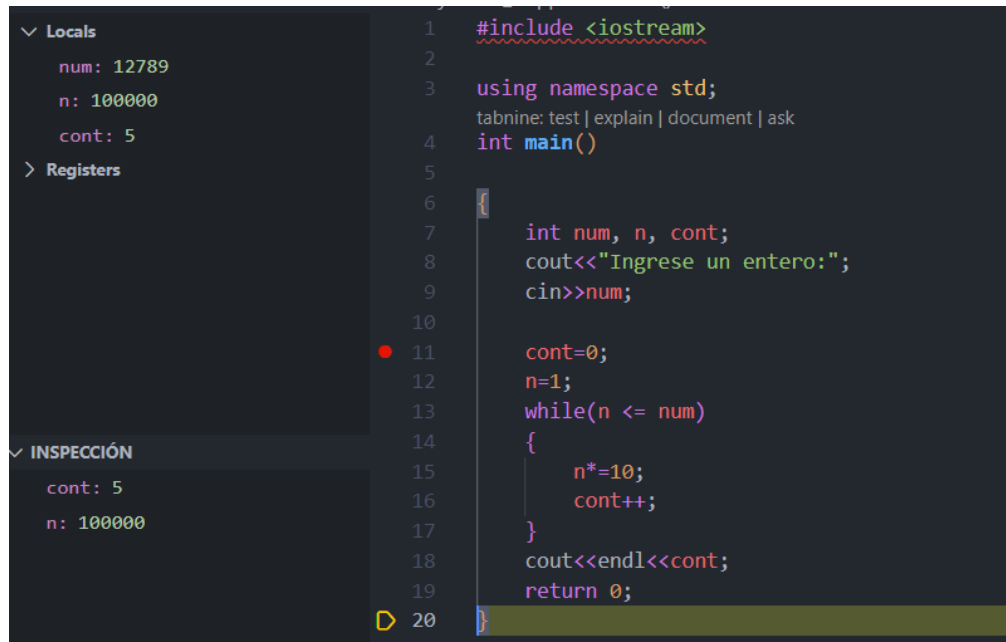
The screenshot shows the same C++ IDE as before, but with the execution point at the start of the while loop. The 'Locals' panel now shows num: 12789, n: 1, and cont: 0. The 'Registers' panel is empty. The 'INSPECCIÓN' panel shows cont: 0 and n: 1. In the code editor, the `while(n <= num)` line is highlighted, and a yellow arrow points to it, indicating the start of the loop.

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```



### Después de salir del bucle cuando el entero es 12789:

Las variables “cont” y “n” después de salir del bucle cuando el entero es 12789, tienen los siguientes valores cont = 5 y la variable n = 100000



```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6
7      int num, n, cont;
8      cout<<"Ingrese un entero:";
9      cin>>num;
10
11     cont=0;
12     n=1;
13     while(n <= num)
14     {
15         n*=10;
16         cont++;
17     }
18     cout<<endl<<cont;
19     return 0;
20 }
```

**Locals**

- num: 12789
- n: 100000
- cont: 5

**Registers**

**INSPECCIÓN**

- cont: 5
- n: 100000