

Práctica 1. Introducción a la programación con C++



(CC) Julio Vega

Realiza los siguientes ejercicios y entrégalos, sin crear ninguna carpeta, bajo el nombre que corresponda, siguiendo la sintaxis `ejercicioX.cpp`; por ejemplo, si es el Ejercicio 1, se debería llamar `ejercicio1.cpp`. En aquellos ejercicios que tengan varios apartados, añade la letra correspondiente al apartado; por ejemplo, en el Ejercicio 2, tendríamos `ejercicio2a.cpp`, entre otros.

Ejercicio 1

Escribe un programa que pida al usuario que escriba dos números, que obtenga los números del usuario e imprima la suma, producto, diferencia y cociente de los números.

Ejercicio 2

Escribe un programa que imprima los números del 1 al 4 en la misma línea, con cada par de números adyacentes separado por un espacio; esto es: 1 2 3 4. Haz esto de varias formas:

- Utilizando una instrucción con un operador de inserción de flujo (`<<`).
- Utilizando una instrucción con cuatro operadores de inserción de flujo.
- Utilizando cuatro instrucciones.

Ejercicio 3

Escribe un programa que pida al usuario que escriba dos enteros, que obtenga los números del usuario e imprima el número más grande, seguido de la expresión *es más grande*. Y, si los números son iguales, que imprima el mensaje *Estos números son iguales*.

Ejercicio 4

Escribe un programa que reciba tres enteros del teclado e imprima la suma, promedio, producto, menor y mayor de esos números. El diálogo de la pantalla debe aparecer de la siguiente manera:

```
Introduzca tres enteros distintos: 13 27 14
```

```
La suma es 54
```

```
El promedio es 18
```

```
El producto es 4914
```

```
El menor es 13
```

```
El mayor es 27
```

Ejercicio 5

Escribe un programa que lea el radio de un círculo como un número entero y que imprima su diámetro, circunferencia y área. Usa el valor constante 3,14159 para π . Realiza todos los cálculos en instrucciones de salida.

Ejercicio 6

Escribe una aplicación que muestre un patrón de tablero de ajedrez como el que se muestra a continuación. Hazlo primero empleando ocho instrucciones de salida (`cout`), y después hazlo intentando utilizar el menor número de instrucciones posible.

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

Ejercicio 7

Escribe un programa que reciba como entrada un número entero de cinco dígitos, que separe ese número en sus dígitos individuales y los imprima, cada uno separado de los demás por tres espacios. Por ejemplo, si el usuario escribe el número 31250, el programa debe imprimir: 3 1 2 5 0.

Ejercicio 8

El factorial de un entero n no negativo se escribe como $n!$ (n factorial) y se define de la siguiente manera: $n! = n(n-1)(n-2)\dots 1$ (para valores de n mayores o iguales a 1) y $n! = 1$ (para $n = 0$ o $n = 1$).

Por ejemplo, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, que es 120. Usa instrucciones **while** en cada uno de los siguientes casos:

- Escribe una aplicación que lea un entero no negativo, que calcule e imprima su factorial.
- Escribe un programa que estime el valor de la constante matemática e , utilizando la fórmula: $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$. Pide al usuario la precisión deseada de e (es decir, el número de términos en la suma).
- Escribe una aplicación que calcule el valor de e^x , utilizando la fórmula: $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$. Pide al usuario la precisión deseada de e (es decir, el número de términos en la suma).

Ejercicio 9

Calcula el valor de π a partir de la serie infinita: $\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$. Imprime una tabla que muestre el valor aproximado de π , después de cada uno de los primeros 1000 términos de esta serie.

Ejercicio 10

Escribe un programa que imprima la siguiente figura de rombo. Puedes utilizar instrucciones de salida que impriman un solo asterisco (*) o un solo espacio en blanco. Maximiza el uso de la repetición (con instrucciones **for** anidadas), y minimiza el número de instrucciones de salida.

```

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

Bonus: haz una mejora del programa (`ejercicio10pro.cpp`) para que lea un número impar en el rango de 1 a 19, correspondiente al número de filas que configuran el rombo, y después muestre un rombo del tamaño apropiado.