

Hallando el valor factorial

Problema:

Realizar un programa que permita hallar el factorial de cualquier número entero positivo, dato ingresado por el usuario comprendido en un rango (0 – 15), este además debe contar con un mensaje emergente que solicite almacenar un dato tipo carácter, escrito por el usuario, con el cual le indique al programa si desea seguir calculando el factorial de otro número dentro del rango o desea terminar la ejecución del programa, es decir debe calcular de forma continua hasta que el usuario indique que ya no desea seguir ejecutando el programa.

Especificaciones:

- El programa deberá terminar su ejecución cuando el usuario lo indique “Antes no”.
- Podrá estimar la cantidad de recursos como variables, mensajes, etc., que requiera necesarios.
- Las operaciones del número factorial deberán aplicarse y detallarse en el desarrollo del programa.
- Se tendrá en cuenta como un adicional en la revisión del programa, el adecuado diálogo con el usuario y el menor uso de condicionales en el programa.

Factorial de un número:

Para hallar el valor factorial de un número, quiere decir que hay que multiplicar todos los números enteros positivos que hay entre ese número y el 1.

Ejemplo: Valor factorial del número 3 y 5

- $¡3 = 3 \cdot 2 \cdot 1 = 6$
- $¡5 = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$

Pista:

Utiliza ciclos o bucles, que son un tipo de condición que ejecuta de forma repetitiva las instrucciones contenidas en ella, hasta que deja de cumplirse la condición que las puso en marcha.

Análisis del problema:

- **Entradas:**

- Dato numérico tipo int (Número) entre el rango [0 - 15]
- Dato tipo char (carácter), el cual suele ser S/N (si/no).

- **Procesos:**

- Calcular el factorial de un número positivo entre [0 - 15] teniendo en cuenta que:

$$n! = \begin{cases} 1 & \text{si, } n = 0 \\ (n - 1)! \times n & \text{si, } n > 0 \end{cases}$$

Donde “n” es el número del que se obtendrá el factorial.

- **Salidas:**

Resultado del factorial del número “n”, detallando el proceso.

Análisis en lenguaje C:

Variables:

El programa obtiene el factorial de números enteros, por lo tanto el resultado siempre será otro número entero. Por lo tanto, se usarán:

- Una variable de tipo INT para almacenar el número que ingresa el usuario.
- Una variable de tipo INT para almacenar el resultado.
- Una variable de tipo INT que será el contador de un ciclo FOR.
- Una variable de tipo INT que será el acumulador de cada multiplicación.

Además, el programa debe ser capaz de salir o terminar su funcionamiento cuando el usuario lo indique, por lo que se usará una variable de tipo carácter que almacenará una letra, por convención sería (S/N), es decir, sí o no. Por lo tanto, se usará:

- Una variable de tipo CHAR para almacenar la decisión del usuario (s/n).

Ciclos o bucles:

Para evitar el uso excesivo de estructuras condicionales se hará uso de ciclos o bucles, por ello es importante saber su funcionamiento.

- En este programa se hará uso del **ciclo Do While**, ya que el programa se ejecutará al menos una vez, y seguirá haciéndolo hasta que el usuario le indique que quiere salir.
- Además, para calcular el factorial del número ingresado, se hará uso de un **ciclo For** ya que para calcular el factorial de un número se debe multiplicar sus antecesores varias veces.

NOTA:

Siguiendo los pasos anteriores, queda claro que se cumplirán las especificaciones que el programa debe tener, así como su correcto funcionamiento ya que:

- La primera especificación indica que el programa debe parar cuando lo desee el usuario, lo cual se consigue con el ciclo Do While y la variable tipo carácter.
- La segunda especificación indica la estimación de variables, memoria, etcétera, que necesitará el programa, lo cual queda definido en este documento.

- La tercera especificación indica que el programa deberá mostrar el proceso que llevó a cabo para obtener el resultado, lo cual se logrará con el ciclo For, la variable acumular, la variable contador y un mensaje.
- La cuarta especificación indica que el programa deberá tener diálogo con el usuario lo cual se logrará con una serie de mensajes, y además, deberá utilizar el menor número posible de estructuras condicionales, lo cual se logrará con los ciclos antes mencionados.

Cabe mencionar que, para un uso de memoria menor, se usará otro tipo de dato que acepta números enteros, el cual se llama Short. (ver la siguiente tabla).

Tipo de Dato	Descripción	Número de bytes típico	Rango
short	Entero corto	2	-32768 a 32767
int	Entero	4	-2147483648 a +2147483647
long	Entero largo	4	-2147483648 a +2147483647
char	Carácter	1	-128 a 127