

Taller 2: Estructuras de control

Haga diagramas de flujo para las siguientes situaciones, tenga presente que el programa debe pedir al usuario información precisa y el resultado o salida del programa debe ser lo mas explícita posible:

1. **Suma de números naturales:** Escribe un programa que solicite al usuario un número natural y luego calcule y muestre la suma de todos los números naturales menores o iguales al número ingresado.
2. **Cálculo del factorial:** Crea un programa que calcule el factorial de un número natural ingresado por el usuario.
3. **Suma de números pares e impares:** Diseña un programa que, dado un número natural ingresado por el usuario, calcule y muestre por separado la suma de todos los números pares y la suma de todos los números impares hasta ese número.
4. **Pregunta de continuación:** Implementa un programa que, utilizando un ciclo `while`, pregunte repetidamente al usuario si desea continuar. El programa debe finalizar solo si el usuario responde exactamente `no`, `No`, o `NO`.
5. **Tabla de multiplicar:** Crea un programa que, al ingresar un número natural entre 1 y 10, muestre la tabla de multiplicar correspondiente.
6. **Divisores de un número:** Escribe un programa que solicite un número natural al usuario y luego muestre todos los divisores de ese número.
7. **Adivina el número:** Desarrolla un programa donde un primer usuario ingrese un número entre 1 y 100, y un segundo usuario intente adivinarlo. El programa debe indicar si el número a adivinar es mayor o menor, y continuar hasta que el segundo usuario adivine correctamente. Mejora el programa para que el número a adivinar sea generado aleatoriamente y el segundo usuario tenga solo 6 intentos para adivinarlo. En cada intento, el programa debe indicar cuántos intentos quedan.

8. **Simulación de una alcancía:** Escribe un programa que simule una alcancía. Primero, el programa debe pedir al usuario que ingrese la cantidad de dinero que desea ahorrar. Luego, debe solicitar repetidamente las cantidades que va ahorrando, hasta que el total ahorrado sea igual o superior al objetivo. El programa solo debe aceptar cantidades positivas.
9. **Máximo común divisor (MCD):** Implementa un programa que calcule el máximo común divisor (MCD) de dos números enteros ingresados por el usuario.
10. **Promedio de números:** Crea un programa que calcule el promedio de una serie de números ingresados por el usuario. El programa debe continuar solicitando números hasta que el usuario ingrese 0, lo cual indicará el final de la entrada de datos. El número 0 no debe incluirse en el cálculo del promedio.
11. **Sucesión numérica** Dada la sucesión de números -3,-1,1,3,5,7,9 ...
 - (a) Hallar el término n -ésimo (a_n) de la sucesión.
 - (b) El programa muestra la cantidad de términos de la sucesión que el usuario desea ver.
 - (c) Determine el cociente entre los dos últimos números de la sucesión $\left(\frac{a_n}{a_{n-1}}\right)$ y determine si este cociente se acerca a algún número cuando la cantidad de términos de la sucesión es grande.
12. * **Sucesión de Fibonacci** Dada la sucesión de números 1,1,2,3,5,8,13...
 - (a) Hallar el término n -ésimo (a_n) de la sucesión.
 - (b) El programa muestra la cantidad de términos de la sucesión que el usuario desea ver.
 - (c) Determine el cociente entre los dos últimos números de la sucesión $\left(\frac{a_n}{a_{n-1}}\right)$ y determine si este cociente se acerca a algún número cuando la cantidad de términos de la sucesión es grande.

13. * **Aproximación de π** A través de la historia, grandes mentes han desarrollado diferentes esquemas de aproximaciones para el cálculo del número π . Consideraremos dos esquemas, uno dado por Leibniz (1646 - 1716), y uno dado por Euler (1707 - 1783).

El esquema dado por Leibniz puede ser escrito como:

$$\pi \approx \sum_{k=0}^N \frac{8}{(4k+1)(4k+3)}.$$

Mientras que el esquema de Euler es:

$$\pi \approx \sqrt{\sum_{k=1}^N \frac{6}{k^2}}.$$

Si solamente los primeros N términos de cada suma son usados como aproximación de π , cada esquema aproximará π con algún error.

Escribir un programa que tome a N como entrada y arroje el valor aproximado de Pi y el error con ambos esquemas. Por último, concluya cuál es el mejor esquema de aproximación.

- (a) * Encuentre el valor de N para el cual el error es menor a 0.0001 en ambos esquemas. Rta: NL=4.999, NE=9.549

*“Si estás trabajando en algo que te importa de verdad,
nadie tiene que empujarte: tu visión te empuja ”
Steve Jobs.*