5)))

¿Para qué situaciones utilizaríamos una estructura de control condicional alternativa? \*

Cuando se requiera que ocurra una de dos situaciones, dependiendo de cierta condición. Al cumplirse dicha condición se ejecutan los comandos dentro del “SI”, o en caso de no cumplirse la condición, la ejecución continúa salteándose el “SI” y en caso de que haya un “SINO” lo ejecuta.

**Si necesitaríamos programar un menú que contiene muchas opciones (más de diez). ¿Sería más óptimo realizar la estructura de control selectiva SEGÚN antes que la estructura de control condicional alternativa? Justificá tu respuesta.**

Si, resulta óptima una estructura de control “SEGÚN” ya que permite anidar muchas estructuras de control “SI” de manera más legible.

**Realizar un algoritmo que permita al usuario ingresar un valor numérico entero. El algoritmo debe informar si el valor ingresado es par o impar.**

El siguiente algoritmo funciona para los números enteros mayores a 0. Es decir, NUM > 0

Definir NUM como Entero

Escribir "Ingrese un número:"

Leer NUM

Mientras (NUM >1)

NUM <- NUM-2

FinMientras

Si (NUM==1)

Escribir "El número es impar."

Sino

Escribir "El número es par."

FinSi

**¿Qué diferencia encontrás entre la estructura MIENTRAS (While) y la estructura HACER-MIENTRAS (Do-While)?**

La estructura While primero verifica que la condición se cumpla, y luego ejecuta las instrucciones si se cumple la condición; cuando la condición deja de ser cumplida, finaliza el bucle. Por otro lado, la estructura Do-While, o Repetir, primero ejecuta una serie de instrucciones y al final verifica si una condición “Hasta que” se cumple; cuando la condición se cumple, finaliza el bucle. Se puede decir que la estructura “Hacer-Mientras” siempre comienza activa y se corta al cumplir la condición, y que la estructura “Mientras” requiere que la condición se cumpla para comenzar; es decir, puede estar activa o inactiva.

¿Qué función cumple la variable del índice en una estructura PARA (For)? \*

La variable del índice en la estructura For sirve como contador que indica cuantas veces debe repetirse el ciclo. El ciclo se ejecuta hasta que el índice llega a su valor final.

Compare características que presenta la estructura PARA (For) con las que presenta la estructura MIENTRAS (While) \*

La estructura For utiliza una variable índice que funciona como contador, debe ser entera y su valor incrementa de a 1 con cada ciclo de ejecución. El bucle se interrumpe cuando el índice llega a su valor final. En la estructura Mientras se pueden trabajar con todos los tipos de variables en la condición. En ambos casos la condición se evalúa antes de ejecutar el ciclo.

**6)))** Explique qué diferencias hay entre un método tipo función y un método tipo procedimiento. ¿Para qué situaciones utilizaría cada uno de los dos? \*

Un módulo tipo función es un conjunto de instrucciones, que siempre requiere un parámetro de entrada que se ingresa entre paréntesis, y siempre devuelve un resultado.

Un módulo tipo procedimiento es también un conjunto de instrucciones que puede devolver un resultado o no. A su vez, puede tener parámetros de entrada o salida, ambos o incluso ninguno.

Una función se podría utilizar para realizar una serie de operaciones matemáticas y obtener un resultado final. Un procedimiento puede dar varios resultados a la vez, por ejemplo ingresar 3 datos y entregar: la suma de ellos, el promedio y cuál es el mayor.

¿Qué beneficios trae aplicar el concepto de modularización en un programa? \*

Descomponer el problema general en subproblemas para simplificarlo. La reutilización de código, que permite agilizar el proceso de desarrollo, y evitar lo que se conoce como “reinventar la rueda”. También facilita el mantenimiento y la prueba de código, al estar aislados los cambios por cada módulo que necesita modificación.

¿Qué diferencia hay entre una variable local y una global? \*

Se diferencian en qué ámbito de memoria son almacenadas, si en la memoria global es decir que se definen en el algoritmo principal, o si en ámbitos de memoria local que se definen dentro de una rutina.

Explique el concepto de “caja negra” en la programación. \*

Se utiliza este concepto para referir a que desconocemos los pasos internos del módulo, pero lo que nos importa son los datos de entradas que necesita y los datos de salida que produce.