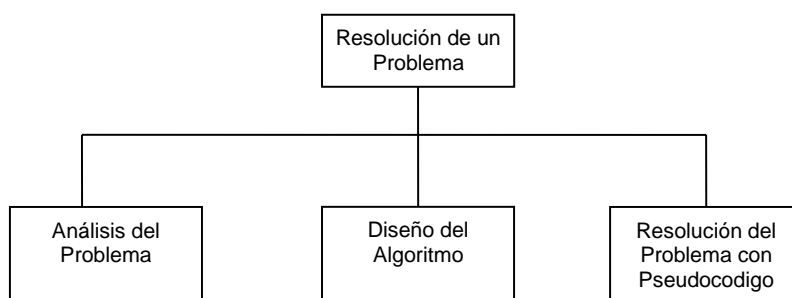


Trabajo Práctico Nº 2

Estructuras de Control de Decisión

Objetivo: Que los estudiantes puedan tomar decisiones en los pseudocódigos que construyen para la resolución de problemas.

La Primera fase de la resolución de un Problema con Pseudocódigo es el Análisis del Problema. Esta fase requiere comprender con claridad el problema, de modo que permita dar una clara definición de todos los aspectos involucrados para de ese modo llegar a una solución. A continuación deben estudiarse las características de esa solución a fin de encontrar la mejor manera de llevarla adelante. De eso se encarga la fase del Diseño, cuyo producto final es el Pseudocódigo. Para obtenerlo, también se precisan especificaciones detalladas de las Entradas y Salidas. Así, las tareas a realizar pueden resumirse en:



En problemas de pequeña o mediana envergadura, lo anterior puede llevarse a cabo respondiendo a las siguientes preguntas:

- ¿Qué Entradas se requieren? (Tipo y Cantidad)
- ¿Cuál es la Salida deseada? (Tipo y Cantidad)
- ¿Qué debe hacerse para lograr la Salida deseada a partir de los datos disponibles?

Ejemplo:

Determinar si la suma de cualquier pareja de tres números dados es igual al tercero. Si se cumple esta condición, escribir "Iguales" y, caso contrario, escribir "Distintos".

Análisis del Problema:

¿Qué Entradas se requieren? (Tipo y Cantidad) Tres números Reales

¿Cuál es la Salida deseada? (Tipo y Cantidad) Un mensaje

¿Qué método produce la Salida deseada? La comparación de la suma de las parejas de números con el número que queda

Diseño de la solución (por refinamientos sucesivos)

Primer intento: a grandes rasgos

*Lo primero que debe notarse al analizar el problema es que será necesario, en algún punto, tomar una decisión y, en consecuencia, mostrar diferentes avisos. Eso nos lleva a pensar en la necesidad de incorporar las estructuras de control **condicionales**, es decir aquellas que nos permiten decidir, por medio de un criterio, cuáles instrucciones deben ejecutarse y cuáles dejarse de lado de entre 2 o más posibles alternativas.*

Inicio

Obtener tres números reales

Calcular las sumas dos en dos de los números obtenidos

Si alguna de ellas es igual al tercer número, mostrar el cartel "Iguales" y terminar; de lo contrario, mostrar el cartel "Distintos" y terminar.

Fin

Segundo intento: buscando los detalles

Al analizar el primer intento, puede interpretarse que habría que calcular todas las sumas primero y luego hacer todas las comparaciones. Sin embargo, esto nos llevaría a un código más extenso y consumidor de recursos, por lo que decidimos explorar otra alternativa: preguntar si se da la igualdad cada vez que calculamos una suma. Así, se obtiene la siguiente propuesta:

```
Inicio
    // Obtener los tres números reales
    Leer A
    Leer B
    Leer C
    // Calcular cada suma y compararla con el tercer número
    Si A+B = C
        Imprimir "Iguales"
    Terminar
    Si A+C = B
        Imprimir "Iguales"
    Terminar
    Si B+C = A
        Imprimir "Iguales"
    Terminar
    Imprimir "Distintas "
    // Terminar
Fin
```

En el pseudocódigo que manejamos, no es posible hacer que un programa termine su ejecución antes de haber llegado al final del mismo, aunque lo que acabamos de escribir da esa idea. Por eso debemos buscar alguna herramienta idónea que nos permita hacer lo que necesitamos respetando las formas establecidas. Eso nos lleva a un nuevo refinamiento.

Tercer intento:

*El programa que se implementa a continuación tiene una típica estructura **secuencial con condiciones**, es decir que consta de instrucciones que se van ejecutando una a continuación de la otra, en el orden en que fueron escritas de arriba hacia abajo, pero sin embargo en algunos puntos debe tomarse una decisión sobre lo que se debe hacer. Debe notarse que la única alteración a la secuencia se da en el sentido de saltar algunas instrucciones.*

Programa En Pseudocódigo:

```
Programa Pares
// Programa escrito por: Gustavo
// Fecha: 26 de Abril de 2009
// Versión: 05
// Nombre del archivo: Pares
// Este programa suma tres números de a pares y compara con el número restante.
// Se utilizan SI anidados.
Variables
A, B, C: Real 5,2
Hacer // Comienzo del programa
// Ingreso de datos
Imprimir: 'Ingrese el primer valor: '
Leer: A
Imprimir: 'Ingrese el segundo valor: '
Leer: B
Imprimir: 'Ingrese el tercer valor: '
Leer: C
// Se aplican SI anidados para comparar
SI (A+B)=C ENTONCES
    Imprimir: ' Son Iguales....! '
SINO
```

```
    SI (A+C)=B ENTONCES
        Imprimir: ' Son Iguales....! '
    SINO
        SI (B+C)=A ENTONCES
            Imprimir: ' Son Iguales....! '
        SINO
            Imprimir: ' Son Distintos...! '
        FIN SI
    FIN SI
FIN SI

Fin Hacer
Fin Programa Pares
```

Observar que en esta versión se anidaron las condiciones de una manera bastante particular: todas las cláusulas **SINO** aparecen alineadas al margen izquierdo, como si todas correspondieran a un mismo **SI**, cuando en realidad cada una de ellas corresponde a un **SI** diferente; esto es porque de ese modo se resalta el hecho de que cada una de las condiciones evaluadas se considera mutuamente excluyente con las demás (de modo que una y solamente una puede ser cierta).

En cada uno de los ejercicios que se plantean a continuación, determinar y explicitar las entradas necesarias, las salidas deseadas y el proceso que las relaciona ANTES de codificar.

Ejercicios sugeridos para trabajar en clase:

1. Si se analiza el pseudocódigo del ejemplo, se destaca que cada una de las condiciones llevaban a hacer exactamente lo mismo en caso de que cualquiera de ellas fuese verdadera, y sólo se hacía algo diferente en el caso en que TODAS ellas resultaban falsas.
Encontrar una manera de construir una condición que permita expresar lo que acabamos de mencionar, de modo que el programa resultante utilice una sola sentencia SI para decidir.
2. Se ingresan 4 números reales y se quiere determinar si la suma de ellos es positiva, negativa o cero. Imprimir un mensaje indicándolo.
3. Se tiene una máquina expendedora de gaseosas cuyo valor es de \$1,25.
Admite monedas de: \$0,10 \$0,25 \$0,50 y \$1 y realiza las siguientes operaciones:
 - a. Venta sin Vuelto
 - b. Venta con Vuelto
 - c. No Venta y devuelve las monedasEl programa debe aceptar las monedas e imprimir un mensaje indicando la operatoria.
4. Un empleado de banco tiene un sistema que imprime un ticket cada vez que un cliente realiza una compra de dólares. La operación tiene una comisión administrativa según la cantidad de dólares que se venden, lo que representa un sobre costo para el cliente:
 - a. 2% si es más de 501 y menos de 1501
 - b. 2,5% si es mayor de 1501 y menos que 3501
 - c. 4 % si es mayor a 3501El ticket muestra discriminados: la cantidad de dólares que se compraron, el valor de cada dólar, el monto de la comisión y el monto total que el cliente debe abonar por la compra.
Hacer un programa que simule la compra e imprima el ticket.
5. Diseñar un algoritmo que calcule el mayor de dos números dados. ¿Cuántas condiciones requiere evaluar su algoritmo? ¿Podría codificar otro que utilice menos? ¿Y otro que utilice menos sentencias SI?
6. ¿Qué errores de sintaxis se detectan en los siguientes grupos de sentencias?

a) **Si** (A > 0) **Entonces**
 Si (B > 0) **Entonces**

C:= A + B

Sino

C:= A + 3 * C

D:= A - B - C

b) **Si** (A > 0)

Imprimir: 'Es Mayor...'

Sino

Imprimir: 'Es menor...'

Si

Imprimir: 'Son Iguales...'

Fin Si

Analizar y mostrar todos los posibles códigos resultantes de intentar corregir el / los error(es) encontrado(s).

7. Realizar un programa que simule una calculadora simple. Lee dos números y un carácter.

Si el carácter es un "+", se imprime la suma; si es un "-", se imprime la diferencia; si es un "*", se imprime el producto; si es un "/", se imprime el cociente (si no puede calcularse debe emitirse un aviso). *Utilizar el CASO.*

Ejercicios sugeridos para trabajar fuera del horario de clase:

8. ¿Qué valores van tomando las variables en este conjunto de instrucciones?

A := 3625,25

B := 0,2568

OP := (A + B) / 152

Imprimir: OP

OP := (OP * A) - B

Imprimir: OP

B := OP * 56

Imprimir: B

Si OP >= 45 **Entonces**

Imprimir: 'Camino 1', A, B

Sino Imprimir: 'Camino 2', A, OP

Fin Si

Imprimir: A, B, OP**3

9. Diseñar un algoritmo que ingrese I, J, K, L. Si I/J=K/L imprimir "Son iguales", en caso contrario no dar mensaje. *Recuerde verificar que los divisores sean distintos de cero y en este caso indicar la imposibilidad de realizar las operaciones.*

10. Cierta atleta pierde en sus carreras 100 gramos de peso por cada 50m recorridos. Su entrenador desea un programa que a partir de datos como

- Peso del corredor
- Longitud del circuito (nunca menor a 50m)
- Tiempo empleado (en seg.)

Cumpla con los siguientes puntos:

a. Determinar cuántos gramos pierde el corredor y cuál es su peso final

b. Si ha perdido más de 300g de peso, indicar que necesitará vitamina A y si pierde más de 450g necesitará también vitamina C

c. Sabiendo que para el circuito olímpico el tiempo récord es de 25 seg., y que para el circuito panamericano el tiempo récord es de 15 seg., determinar si el corredor ha mejorado alguno de estos récords.

11. Diseñar un algoritmo que decida cuántas de las soluciones de la ecuación $ax^2 + bx + c$ son números reales. (*Sugerencia: recordar que la cantidad de soluciones sólo puede ser 0, 1 ó 2, y que se obtienen de la fórmula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$*).
12. Sabiendo la cantidad de bancos de un aula y la cantidad de alumnos inscriptos para ese curso, determinar si alcanzan los bancos existentes. De no ser así informar además cuantos bancos es necesario agregar.
13. Diseñar un algoritmo que calcule el mayor de 3 números dados. ¿Cuántas condiciones requiere evaluar su algoritmo? ¿Podría codificar otro que utilice menos? ¿Y otro que utilice menos sentencias **SI**?
14. Diseñar un algoritmo que dados dos puntos del plano P y Q imprima las coordenadas del vector PQ y las del vector QP .