

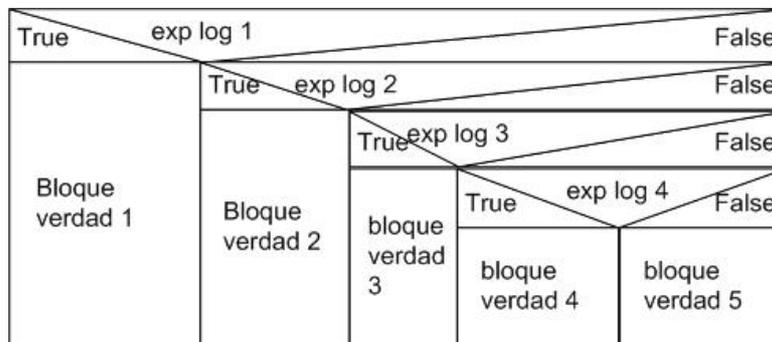
CONDICIONALES MÚLTIPLES

Cuando se desea utilizar un condicional que no tenga solo dos si no más alternativas (excluyentes entre sí) es posible utilizar la instrucción ELSE IF (De lo contrario Si) en vez de utilizar IF anidados. Esto para que el algoritmo y el código sean más claros.

Ejemplo: Diseñar un algoritmo para leer la nota final de un estudiante y decir si el estudiante ganó, perdió o puede habilitar.

IF anidado	ELSE IF
<i>INICIO</i> Leer Nota Si Nota >= 3 Imprimir "Ganó" De lo contrario Si Nota >= 2 Imprimir "Puede habilitar" De lo contrario Imprimir "Perdió" Fin Si Fin Si <i>FIN</i>	<i>INICIO</i> Leer Nota Si Nota >= 3 Imprimir "Ganó" De lo contrario Si Nota >= 2 Imprimir "Puede habilitar" De lo contrario Imprimir "Perdió" Fin Si <i>FIN</i>

Utilizar ELSE IF es útil cuando se encuentran varios IF anidados y por la parte verdadera se llega a una respuesta, mientras que por la falsa se debe seguir preguntando, tal como se presenta a continuación:



Al representar el ELSE IF mediante un diagrama de caja o un diagrama de flujo, queda igual que un IF anidado pues conceptualmente son iguales. Esto significa que la contracción de estas instrucciones sólo se percibe en pseudocódigo y en código.

Al codificar en el lenguaje C la segunda alternativa (utilizando ELSE IF) quedaría:

```
#include <iostream.h>
```

```
void main()
{
    float Nota;
```

```

cout<<"Ingrese la nota final del estudiante :";
cin>>Nota;

if (Nota >= 3)
    cout<<"\nGanó";
else if (Nota >= 2) //si no es mayor o igual a 3 pero si que 2
    cout<<"\nPuede habilitar";
else //si no es ninguna de las anteriores (entonces es menor que 2)
    cout<<"\nPerdio";

}

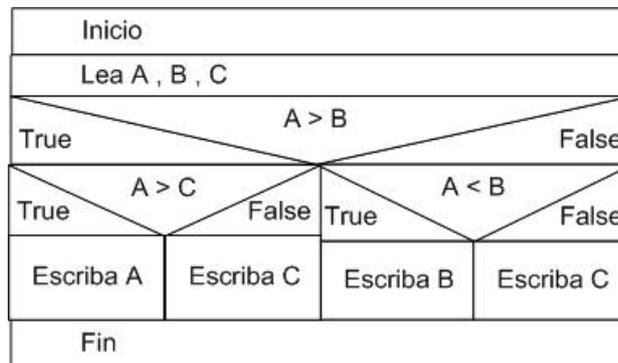
```

Nótese que al usar ELSE IF es como si se utilizaran dos instrucciones a la vez (ELSE e IF), lo que ahorra llaves en C.

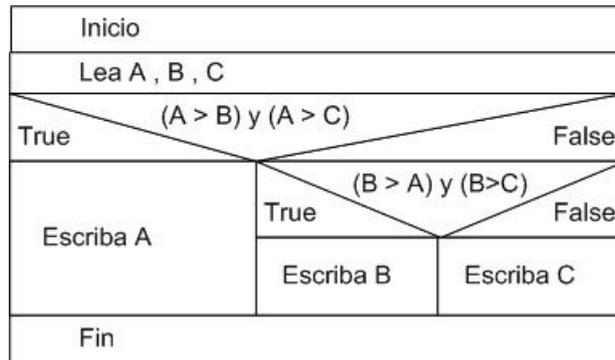
EJERCICIOS

1. Diseñar un algoritmo que lea tres valores numéricos y escriba el mayor valor ingresado

Solución en diagrama de caja:



Importante: Recuerden que no existe una manera única de resolver un problema, por lo que pueden existir diversos algoritmos que hagan lo mismo. A alguien por ejemplo se le pudo haber ocurrido la solución del problema anterior de la siguiente manera (siendo también válida):



Al pasar el algoritmo anterior en C se puede utilizar tanto IF anidado como ELSE IF, como se muestra a continuación.

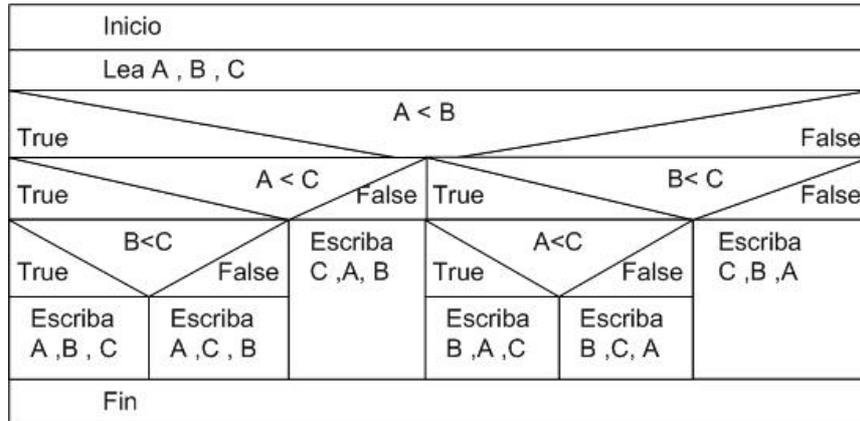
IF anidado	ELSE IF
<pre>#include <iostream.h> void main() { float A, B, C; cout<<"Ingrese A: "; cin>>A; cout<<"\nIngrese B: "; cin>>B; cout<<"\nIngrese C: "; cin>>C; if (A > B) && (A > C) { cout<<"\n"<<A; } else { if (B > A) && (B > C) { cout<<"\n"<<B; } else { cout<<"\n"<<C; } } }</pre>	<pre>#include <iostream.h> void main() { float A, B, C; cout<<"Ingrese A: "; cin>>A; cout<<"\nIngrese B: "; cin>>B; cout<<"\nIngrese C: "; cin>>C; if (A > B) && (A > C) { cout<<"\n"<<A; } else if (B > A) && (B > C) { cout<<"\n"<<B; } else { cout<<"\n"<<C; } }</pre>

2. Este ejercicio es diferente al anterior. También se leen tres valores numéricos A, B y C pero lo que el algoritmo debe hacer es escribirlos en orden ascendente. Por ejemplo, si se ingresan los números 3 1 2, debería imprimir 1 2 3.

Solución: Con 3 números A, B y C cualesquiera las formas en que pueden organizarse para quedar ordenadas son:

ABC, ACB, CAB, BAC, BCA y CBA, es decir, todas sus posibles combinaciones.

Al diseñar el algoritmo en diagrama de caja puede quedar como se presenta a continuación:



Al pasar este algoritmo a pseudocódigo usando IF anidados queda:

INICIO

```

Leer A, B, C
Si A < B
    Si A < C
        Si B < C
            Escriba A, B, C
        En caso contrario
            Escriba A, C, B
    Fin Si
    En caso contrario
        Escriba C, A, B
    Fin Si
En caso contrario
    Si B < C
        Si A < C
            Escriba B, A, C
        En caso contrario
            Escriba B, C, A
        Fin Si
    En caso contrario
        Escriba C, B, A
    Fin Si
Fin Si
Fin Si
FIN

```

Cómo podría reformularse este algoritmo si no se usaran IF anidados si no ELSE IF con preguntas compuestas?

3. Diseñar un algoritmo que lea tres valores numéricos L1, L2, L3 y averigüe si forman un triángulo. En caso positivo decir si es equilátero, isósceles o escaleno. Para que tres valores formen un triángulo se debe comprobar que ningún lado sea mayor que la suma de los otros dos.

Solución: Para resolver este problema primero se deben leer los tres valores y determinar si pueden conformar un triángulo. Al diseñar esta parte en pseudocódigo queda:

INICIO

Leer L1, L2, L3

Si ((L1 < L2 + L3) Y (L2 < L1 + L3) Y (L3 < L1 + L2))

Escribir "Forman un triángulo."

Si ((L1 = L2) Y (L2 = L3))

Escribir "El triángulo es equilátero"

De lo contrario Si ((L1 = L2) O (L2 = L3) O (L1 = L3))

Escribir "El triángulo es isósceles"

De lo contrario

Escribir "El triángulo es escaleno"

Fin Si

De lo contrario

Escribir "No Forman un triángulo"

Fin Si

FIN

El estudiante debe traducir este pseudocódigo a código en C.