

## PRACTICO N° 3

El objetivo de este práctico es ejercitar el uso de las estructuras de control y determinar la estructura más adecuada en cada caso. **RECOMENDACION:** 1) analizar en papel los ejercicios y 2) verificar en la computadora. Probar todos los ejercicios en el entorno Octave. Para la elección de la estructura, en los ejercicios de iteración, se debe procesar solamente los elementos indispensables para devolver el resultado.

Se desestimula el uso del comando **disp** al programar. En este práctico, **disp** se usa PERO solamente como apoyo para entender las estructuras de control y selección. **Observar** que los **scripts** aquí presentados, son simples ejemplos de instrucciones de iteración, secuencia y selección.

### Ejercicio 1

El siguiente código es un ejemplo de *script*, y se puede escribir de muchas formas diferentes a la presentada. Escriba por lo menos dos formas distintas a la presentada: una utilizando **elseif** y otra utilizando **switch-case**.

```
if Mes==1
    y= 'Mes de Enero';
end
if Mes==2
    y= 'Mes de Febrero';
end
if Mes==3
    y= 'Mes de Marzo';
end
disp(y)
```

### Ejercicio 2

Para cada uno de los siguientes problemas, decidir cuál de las instrucciones de selección (**if-else** o **if**) es más apropiada y escribir la porción de código correspondiente.

- Los estudiantes que son candidatos para ser admitidos a un colegio mandan sus puntuaciones. Si una puntuación de un estudiante, es mayor o igual a un cierto valor mínimo, entonces se imprime una carta de aceptación del estudiante. Si no, se envía una notificación de rechazo.
- Para los empleados que hayan trabajado más de 40 horas a la semana. Hay que calcular la paga extra y añadirlo a la paga regular.
- Al resolver una ecuación cuadrática, siempre que el valor del discriminante ( $b^2-4ac$ ) sea negativo, la computadora debe imprimir un mensaje que notifique que las raíces son imaginarias.
- En un aserradero controlado por computadora, si la sección de cruce de un tronco es mayor que ciertas dimensiones, la sierra ha de ajustarse para cortar 4x8 dientes, si no, se ajusta para cortar 2x4.

### Ejercicio 3

Indique qué se muestra **en pantalla**, al ejecutar los siguientes *scripts*:

<pre>a)  x = 1;     y = 2;     z = 3;     w = 0;     if x &gt; y         if y &gt; z             if x &gt; z                 w= x;             else                 w=y;             end         else             w=z         end     end     disp(w)</pre>	<pre>b)  a = 0;     b = -1;     if a &gt; 0         z= 'A'     else         if b &lt; 0             z= 'B'         else             z= 'C'         end     end     disp(z)</pre>
---	--

## Ejercicio 4

Considere la siguiente proposición **if** anidada en el *script* que se plantea a continuación:

```
if a > b
    if a > c
        disp('A es el grande')
    end
end
```

Escriba una sola proposición **if** que sea equivalente a ésta pero que no esté anidada.

## Ejercicio 5

Se desea construir un script que permita clasificar números reales, diciendo si pertenecen, o no, a los siguientes intervalos:

Intervalo 1:  $[-1, 1]$   
Intervalo 2:  $[5, 7]$   
Intervalo 3:  $[0, 1]$   
Intervalo 4:  $[3, 6]$

- a) Para esto, el script debe devolver el/los valores de intervalo/s al cual pertenece el número de entrada. En caso de no pertenecer a ningún grupo, la solución debe de ser vacía.

Para resolver este problema, se propone el siguiente código:

```
% Se supone que x está en la memoria
s = [];
if ((x >= -1) & (x <= 1))
    s = [s 1];
elseif ((x >= 5) & (x <= 7))
    s = [s 2];
elseif ((x >= 0) & (x <= 1))
    s = [s 3];
elseif ((x >= 3) & (x <= 6))
    s = [s 4];
end
```

Ejecute el código propuesto en Octave para los siguientes valores numéricos:

- 2
- 3
- 0.5
- 5.7
- -4
- 6

¿Qué sucede con los resultados?

- b) Se propone también el siguiente código:

```
% Se supone que x está en la memoria
s = [];
if ((x >= 3) & (x <= 6))
    s = [s 4];
elseif ((x >= 5) & (x <= 7))
    s = [s 2];
elseif ((x >= 0) & (x <= 1))
    s = [s 3];
elseif ((x >= -1) & (x <= 1))
    s = [s 1];
end
```

Ejecute el código propuesto en octave para los valores numéricos de la parte a)

¿Qué sucede con los resultados?

- c) De ser necesario, reescriba el código para que cumpla con lo que se pide en la letra.

## Ejercicio 6

¿Cuáles son los contenidos de *suma* y *numero* al final de la ejecución del siguiente script?

```
v=[5,6,-3,0,7,-4,5,8,9];
n=8;
suma=0;
i=1;
indicador=0;

while (i<=n) & ~indicador
    numero = v(i);
    if numero > 0
        suma=suma+numero;
    else
        if numero==0
            indicador = 1;
        end
    end
    i=i+1;
end
disp(suma); disp(numero);
```

## Ejercicio 7

Determinar qué se muestra en pantalla, al ejecutar los siguientes segmentos de scripts:

a)	<pre>for b = 1:3     if b &lt;= 1         a = b - 1;     end     if b &lt;= 2         a = a - 1;     else         a = a + 1;     end     disp(a); end</pre>	b)	<pre>bajo = 1; for k = bajo:3     bajo = bajo + 2;     disp(k)     disp(bajo) end</pre>	c)	<pre>alto = 4; for k = alto:-1:3     disp(k)     disp(alto) end</pre>
----	---	----	---	----	---

## Ejercicio 8

Escribir un script en Octave que despliegue en pantalla los números del 1 al 100 salvo que:

- el número sea múltiplo de 3; en este caso deberá imprimir “OSCAR” en lugar del número.
- el número sea múltiplo de 5; en este caso deberá imprimir “MAGGIOLO” en lugar del número.

**Aclaración:** si el número es múltiplo de 3 y 5, se deberán imprimir los dos textos en lugar del número.

## Ejercicio 9

Revise la estructura de control del siguiente programa y reescríbalo según los criterios presentados en Computación1:

```
suma=0;
cont=1;
while cont<=1000
    suma=suma+cont;
    cont=cont+1;
end
```

## Ejercicio 10

Antes de hacer el ejercicio, lea en el "help" la descripción sobre las funciones *sprintf* y *fprintf*. Luego escriba el siguiente **script** Octave en un archivo llamado **tabla\_verdad\_AND.m**

Recordar que "disp" se usa en este programa, para que Ud vea la salida de la función *sprintf*.. Cuando programa NO use disp, SALVO que quiera depurar el código (*debuggear*).

```
disp(sprintf('x y | x & y\n'));
for i=0:1
    for j=0:1
        disp(sprintf('%i %i | %i\n', i, j, i & j));
    end
end
```

En Octave, defina como "directorio de trabajo" el que usó para salvar el archivo "tabla\_verdad\_AND.m" y luego,

a) Ejecute el siguiente comando:

```
>> tabla_verdad_AND
```

La salida obtenida debería ser similar a la siguiente:

```
x y | x & y
0 0 | 0
0 1 | 0
1 0 | 0
1 1 | 1
```

Este script, tal como su nombre lo indica, imprime en pantalla la tabla de verdad del operador AND.

b) Escriba ahora "**tabla\_verdad\_OR**" y "**tabla\_verdad\_NOT**", basándose en la versión "AND". **OBSERVAR** que el operador NOT es unario y que los operadores AND y OR son binarios.

## Ejercicio 11

a) ¿Qué mostrará el siguiente script, siendo el valor de la variable n igual a 11?

```
numero=1;
v=[];
suma=0;
while numero < n & suma <1000
    numero = numero + 1;
    v=[v,numero];
    suma=suma+numero;
end
disp(v)
```

b) Cambiando simplemente el orden de las sentencias hacer que el bucle devuelva un vector v con los números del 1 al 10.

## Ejercicio 12

Escriba una instrucción *while* que se detenga cuando el valor de la variable booleana **hecho** se haga verdadero. El valor de esta variable se inicializa en falso y se actualiza con el resultado de llamar a la función **hacer\_algo()**.

## Ejercicio 13

Para cada uno de los casos escribir y probar en Octave.

- a) un *script* que cuente las veces que el número  $x$  que se encuentra en un vector  $v$ , de largo 100.
- b) un *script* que encuentre el valor mínimo de los números contenidos en un vector dado  $v$ .
- c) un *script* que cuente y sume los primeros 10 números pares de un vector  $v$ .
- d) un *script* que devuelva el valor mínimo de un vector  $v$  que contiene números enteros no negativos.

**Observar** que los números en  $v$ , son mayores o iguales a 0. Utilizar esta propiedad para seleccionar la estructura adecuada.

## Ejercicio 14

El costo de enviar por correo una carta es de \$ 22 para cartas que pesan hasta 30 grs (inclusive) y \$ 22 más \$ 17 por cada 30 grs adicionales o fracción para cartas que pesan más de 30 grs. Dado el peso de una carta como un número real en gramos, escribir un script "*calcula\_costo*" que despliegue en pantalla el costo de enviarla.

**Ejemplo** en consola:

```
>>peso=31;  
>>calcula_costo  
>>costo_envio=  
39
```

## Ejercicio 15

El isótopo radiactivo plutonio-235 tiene una vida media de 26 minutos. La vida media es el tiempo que se necesita para que se degenere la mitad de la masa del isótopo. Por tanto, después de 26 minutos quedará la mitad, después de otros 26 minutos, quedará la mitad de esta mitad, o sea un cuarto de la cantidad inicial de masa. Escriba una función de Octave, utilizando estructuras de control, que determine en cuánto tiempo el plutonio-235 se degenerará hasta quedar un porcentaje determinado (ejemplo 16%) de su masa inicial, (pasado como parámetro).

## Ejercicio 16

El siguiente código posee un error. Si bien funciona correctamente en Octave, no necesariamente lo hará en otros lenguajes o versiones del mismo programa. Encuentre el error y escriba otra versión que corrija el error encontrado.

```
a = 1;  
ppar = 1;  
for i=1:8  
    if i/2==round(i/2)  
        a = a+1;  
    else  
        i = i+a;  
    end  
    ppar = ppar*a;  
end
```

## Ejercicio 17

Los resultados de aplicar el siguiente script "factorial de n", no son los esperados; reescríbalo. Suponga que  $n$  está en la memoria de trabajo.

```
res = 1;  
while i>=2  
    res = res*i;  
end
```