

PRACTICO N° 5

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.

```

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

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.

- a) 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.
- b) 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.
- c) Se tiene las variables a, b y c, siempre que el valor de (b^2-4ac) sea negativo, la computadora debe imprimir un mensaje que notifique que las raíces son imaginarias.
- d) 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*:

<p>a)</p> <pre> x = 1; y = 2; z = 3; w = 0; if x > y if y > z if x > z w= x; else w= y; end else w= z; end end </pre>	<p>b)</p> <pre> a = 0; b = -1; if a > 0 z='A'; else if b < 0 z= 'B'; else z= 'C'; end end </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
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 número/s 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 para x :

- 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==0
    numero = v(i);
    if numero > 0
        suma=suma+numero;
    else
        if numero==0
            indicador = 1;
        end
    end
    i=i+1;
end
```

Ejercicio 7

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

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

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 “ELADIO” en lugar del número.
- el número sea múltiplo de 5; en este caso deberá imprimir “DIESTE” 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 el curso:

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

Ejercicio 10

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

```

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

Ejercicio 11

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

- un *script* que cuente las veces que el número x que se encuentra en un vector v , de largo 100.
- un *script* que encuentre el valor mínimo de los números contenidos en un vector dado v .
- un *script* que cuente y sume los primeros 10 números pares de un vector v .
- 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 12

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* “*calculo_costo*” que despliegue en pantalla el costo de enviarla.

Ejemplo en consola:

```

>>peso=31;
>>calculo_costo
>>costo_envio=
    39

```

Ejercicio 13

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 14

Los resultados de aplicar el siguiente *script* "factorial de n ", no son los esperados; reescríbalo. Suponga que n está en la memoria de trabajo. El factorial de n es: $n! = n*(n-1)*(n-2)*...*2*1$.

```

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

```

Ejercicio Planilla de cálculo 1

- e) Abra el documento **planilla1.ods** (archivo que fue utilizado en el ejercicio 1 del repartido 2);
- f) mediante la función **=SI** (=IF si Calc está en inglés) agregue una columna que diga para cada alumno si, a partir del puntaje obtenido entre los parciales (la suma de los dos parciales), debe recursar (o no) la asignatura;
- g) inserte otra columna en la cual se pueda observar el resultado del curso: **EXONERA**, (60 o más puntos) gana el derecho a **EXAMEN** (25 o más puntos) o **RECURSA** (cualquier otro caso);
- h) inserte un filtro a la columna (**DATOS-> FILTRO AUTOMÁTICO, DATA → AUTOMATIC FILTER**).

Ejercicio Planilla de cálculo 2

Considere la hoja *Si* de la planilla Rangos_y_Si.ods.
Resuelva utilizando la función SI:

Caso 1: Devuelve “menor” si el valor1 es menor que el valor2, en caso contrario devuelve "mayor o igual".

Caso 2: Devuelve “iguales” si el valor1 es igual que el valor2, en caso contrario devuelve “distintos”.

Caso 3: Devuelve “mayor” si el valor1 es mayor que el valor2, en caso contrario devuelve “menor o igual”.

Caso 4: Devuelve “hay iguales”, si al menos hay dos valores iguales entre valor1, valor2 y valor3. En caso contrario devuelve “no hay iguales”.

Caso 5: Devuelve “correcto”, si valor1 es menor que valor2 y valor2 es menor que valor3. En caso contrario devuelve incorrecto.

Para los siguientes casos deberá usar más de una función SI (en general anidando los SI):

Caso 6: Devuelve “primero” si valor1 es el más chico de los tres. Devuelve “segundo” si el más chico de los tres es valor2. En caso contrario, devuelve “tercero”.

Caso 7: Devuelve el valor más grande entre valor1, valor2 y valor3.

En algunos casos existe la posibilidad de utilizar las funciones Y u O para tener menos Si anidados.

Caso 8: Devuelve el valor más grande entre valor1, valor2 y valor3 utilizando un solo SI.

Algunas preguntas de examen teórico

Hay una sola opción correcta para cada pregunta

Una estructura de control:

- a. es un mecanismo que permite modificar el flujo de ejecución de un programa
- b. son los datos que son manipulados por un programa (variables y constantes)
- c. es toda instrucción contenida dentro de un programa
- d. ninguna respuesta es correcta

Las estructuras de control **IF – ELSE** y **WHILE** tienen en común:

- a. ser estructuras de repetición
- b. ser estructuras de selección
- c. la evaluación de expresiones lógicas
- d. ninguna respuesta es correcta

Una variable es:

- a. un lugar de memoria cuyo valor y nombre pueden ser modificados durante la ejecución del programa
- b. un lugar de memoria, cuyo valor no puede ser modificado, pero su nombre sí, durante la ejecución del programa
- c. un lugar de memoria cuyo valor puede ser modificado, pero su nombre no, durante la ejecución del programa
- d. ninguna respuesta es correcta

¿Qué instrucción utilizaría para cumplir con el siguiente requerimiento?

“Si se cumple una determinada condición C debe ejecutarse un grupo de instrucciones A, pero si la condición no se cumple debe ejecutarse otro grupo de instrucciones B, distinto del anterior”.

- a. FOR
- b. IF-ELSE
- c. WHILE
- d. Ninguna respuesta es correcta