



## ACTIVIDADES PRÁCTICAS 3

### Estructuras de tipo condicional y de selección

---

- 1) Diseñar un algoritmo que pida por teclado tres números; si el primero es negativo, debe imprimir el producto de los tres y si no lo es, imprimirá la suma.
- 2) Crear un algoritmo (y su correspondiente diagrama de flujo) que pida al usuario un número y le diga si es positivo, negativo o cero.
- 3) Crear un Algoritmo que lea tres números distintos y nos diga cuál de ellos es el mayor (recuerda usar la estructura condicional Si y los operadores lógicos).
- 4) Realizar un algoritmo que lea un número por teclado. En caso de que ese número sea 0 o menor que 0, se saldrá del programa imprimiendo antes un mensaje de error. Si es mayor que 0, se deberá calcular su cuadrado y la raíz cuadrada del mismo, visualizando el número que ha tecleado el usuario y su resultado ("Del número X, su potencia es X y su raíz X"). Para calcular la raíz cuadrada se puede usar la función interna RAIZ(X) o con una potencia de 0,5.
- 5) Una tienda ofrece un descuento del 15% sobre el total de la compra durante el mes de octubre. Dado un mes y un importe, calcular cuál es la cantidad que se debe cobrar al cliente.
- 6) Realizar un algoritmo (y su correspondiente diagrama de flujo) que dado un número entero, visualice en pantalla si es par o impar (para que un número sea par, se debe dividir por dos y su resto debe ser 0).
- 7) Crear un Algoritmo que dado un año, nos diga si es bisiesto o no. Un año es bisiesto bajo las siguientes condiciones: Un año divisible por 4 es bisiesto y no debe ser divisible entre 100. Si un año es divisible entre 100 y además es divisible entre 400, también resulta bisiesto.
- 8) Escribir un código que categorice la calificación ingresada por un docente de la siguiente forma (utilice una estructura de selección):  
  
9-10: Ha obtenido una calificación sobresaliente  
  
7-8: Ha obtenido una calificación notable  
  
4-6: Ha obtenido un aprobado  
  
1-3: Ha desaprobado  
  
Validar que la nota ingresada se encuentre entre 1 y 10.
- 9) Crear un programa que pida un número al usuario un número de mes (por ejemplo, el 4) y escriba el nombre del mes correspondiente (por ejemplo, "abril"). Validar que el mes se encuentre entre 1 y 12.



- 10)** Desarrollar un programa que permita elegir qué salida mostrar en pantalla de acuerdo a qué opción el usuario selecciona mediante un menú. Deberá desplegar el siguiente menú:

Menú de recomendaciones

1. Literatura
2. Cine
3. Música
4. Videojuegos
5. Salir

Dependiendo de la opción seleccionada deberá mostrar lo siguiente:

Lecturas recomendables:

- + Esperándolo a Tito y otros cuentos de fútbol (Eduardo Sacheri)
- + El juego de Ender (Orson Scott Card)
- + El sueño de los héroes (Adolfo Bioy Casares)

Películas recomendables:

- + Matrix (1999)
- + El último samurai (2003)
- + Cars (2006)

Discos recomendables:

- + Despedazado por mil partes (La Renga, 1996)
- + Bufalo (La Mississippi, 2008)
- + Gaia (Mägo de Oz, 2003)

Videojuegos clásicos recomendables:

- + Día del tentáculo (LucasArts, 1993)
- + Terminal Velocity (Terminal Reality/3D Realms, 1995)
- + Death Rally (Remedy/Apogee, 1996)

Mensaje de salida:

- + Adiós!

En caso de no seleccionar alguna de las opciones válidas, mostrar el mensaje "Opción no válida". Luego de mostrar alguna de las opciones



elegidas deberá esperar a que el usuario ingrese una tecla. Utilice las funciones “Limpiar Pantalla” y “Esperar tecla” de PSeInt para que el menú siempre aparezca en la parte superior de la pantalla. Para el ejercicio utilizar la estructuras Repetir-Hasta Que y Según.

### Estructuras de iteración controladas por contador

---

- 11) Crear un algoritmo que escriba "Hola" cinco veces.
- 12) Crear un algoritmo que pida al usuario 5 datos y muestre su suma.
- 13) Desarrollar un algoritmo que, dados dos números naturales **b** y **e**, la computadora muestre el valor de **b<sup>e</sup>**, sin utilizar operadores o funciones que calculen exponentes. **Ejemplo: Se ingresa 2 y 5, la computadora muestra: 32.**
- 14) Crear un algoritmo que pida al usuario un número y un símbolo, y dibuje un cuadrado usando ese símbolo. El cuadrado tendrá el tamaño que ha indicado el usuario. Por ejemplo, si el usuario introduce 4 como tamaño y \* como símbolo, deberá escribirse algo como:  
  
\*\*\*\*  
  
\*\*\*\*  
  
\*\*\*\*  
  
\*\*\*\*
- 15) Se pide representar el algoritmo que nos calcule la suma de los N primeros números naturales. N se leerá por teclado. Ejemplo, si ingresamos 4, hacer:  $1+2+3+4 = 10$ .
- 16) Crear un algoritmo (y su correspondiente diagrama de flujo) que visualice la cuenta de los números que son múltiplos de 2 o de 3 que hay entre 1 y 100.
- 17) Desarrollar un algoritmo que permita ingresar un número **N**. Acto seguido, permitir ingresar **N** números. La computadora muestra cuál fue el mayor y en qué orden apareció. **Ejemplo: Se ingresa 5, luego 4 8 6 7 5, la computadora muestra: "El mayor es el 8 en la 2º posición".**
- 18) Desarrollar un algoritmo que permita ingresar un número natural. La computadora muestra el factorial del número. **Ejemplo: Se ingresa 5, la computadora muestra: 120.**
- 19) Crear un algoritmo que lea un número entero (lado) y a partir de él cree un cuadrado de asteriscos con ese tamaño. Los asteriscos sólo se verán en el borde del cuadrado, no en el interior. Ejemplo, para lado = 4 escribiría:



\*\*\*\*

\* \*

\* \*

\*\*\*\*

- 20) Crear un algoritmo que lea un número entero (altura) y a partir de él cree una escalera invertida de asteriscos con esa altura. Deberá quedar así, si ponemos una altura de 5.

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

### Estructuras de iteración controladas por bandera

---

- 21) Crear un algoritmo que calcule la raíz cuadrada del número que introduzca el usuario. Si se introduce un número negativo, debe mostrar un mensaje de error y volver a pedirlo (tantas veces como sea necesario).
- 22) Teniendo en cuenta que la clave es “eureka”, escribir un algoritmo que nos pida una clave. Solo tenemos 3 intentos para acertar, si fallamos los 3 intentos nos mostrará un mensaje indicándonos que hemos agotado esos 3 intentos. Si acertamos la clave, saldremos directamente del programa.
- 23) Crear un algoritmo (y su correspondiente diagrama de flujo) que lea números enteros hasta teclear 0, y nos muestre el máximo, el mínimo (sin considerar el 0) y la media (promedio) de todos ellos. Piensa cómo debemos inicializar las variables.
- 24) Desarrollar un algoritmo que pida un número al usuario y que por cada carga pregunte si se desea seguir ingresando, de la siguiente forma "**¿Desea ingresar otro número? [S/N]**". La carga de datos finaliza cuando se detecta una 'n' o 'N'. La computadora debe mostrar la cantidad de números pares ingresados.
- 25) Realizar un programa que pida al usuario una clave de acceso ("1234"), y que no le permita seguir hasta que la introduzca correctamente.
- 26) Crea un programa (y su correspondiente diagrama de flujo) que pida al usuario un usuario y una contraseña. Deberá repetirse hasta que el usuario sea "admin" y la contraseña sea "1234".



- 27)** Hacer un programa que permita calcular la suma de pares de números. Pedirá dos números al usuario y mostrará su suma, volviendo a repetir hasta que ambos números introducidos sean 0. Deberás usar “Repetir-Mientras”.
- 28)** Preparar un programa que divida dos números que introduzca el usuario. Si el segundo número es cero, se le deberá avisar y volver a pedir tantas veces como sea necesario, hasta que introduzca un número distinto de cero, momento en que se calculará y mostrará el resultado de la división.
- 29)** Leer tres números que denoten una fecha (día, mes, año). Comprobar que es una fecha válida. Si no es válida escribir un mensaje de error. Si es válida escribir la fecha cambiando el número del mes por su nombre. Ej. si se introduce 1 2 2006, se deberá imprimir “1 de febrero de 2006”. El año debe ser mayor que 0. Utilice la estructura “Según”. Nota: en PSeInt, si queremos escribir sin que haya saltos de línea, al final de la operación escribir escribimos sin saltar.
- 30)** Calcular las calificaciones de un grupo de alumnos. La nota final de cada alumno se calcula según el siguiente criterio: la parte práctica vale el 10%; la parte de problemas vale el 50% y la parte teórica el 40%. El algoritmo leerá el nombre del alumno, las tres notas, escribirá el resultado y volverá a pedir los datos del siguiente alumno hasta que el nombre sea una cadena vacía. Las notas deben estar entre 0 y 10, si no lo están, no imprimir