

Ejercicios de programación de Algoritmos.

Instrucciones:

1. Analizar el problema propuesto.
2. Confeccionar digrama de flujo previsto.
3. Realizar el algoritmo en el programa PSeInt
4. Compara ambos diagramas.
5. Revisa que la secuenciación de instrucciones se realiza de forma óptima y el resultado es el deseado.

<http://pseint.sourceforge.net/index.php?page=pseudocodigo.php>

1. ALGORITMOS CON OPERADORES NÚMERICOS.

1.1.- Realizar un algoritmo que asigne un valor a las variables “Num1” y “Num2” y a continuación calcule su suma, su resta, su producto y su división, mostrando los resultados por pantalla. Ejemplo:

Num1 <- 10.

Num2 <- 5

Suma: 15; (Num1 + Num2)

Resta: 5; (Num1 - Num2)

Multiplicación: 50; (Num1 * Num2)

División: 2; (Num1 / Num2)

1.1.1 - Modifica el algoritmo anterior para que los valores asignados a las variables “Num1” y “Num2” sean solicitados e insertados por el usuario por línea de comando.

Num1 <- Leer valor 1

Num2 <- Leer valor 2

1.2.- Realizar un algoritmo que lea dos variables numéricas que el usuario teclee y a continuación cambiar el valor de las variables. Ejemplo si el usuario teclea:

Num1 = 10;

Num2 = 20;

El resultado del algoritmo debería de ser:

Num1=20;

Num2=10;

1.3.- Diseñar un algoritmo al cual se le pasará la cantidad de alumnos chicos y alumnos chicas de un centro, con el fin de calcular el % de chicos y el % de chicas del mismo. Pista: el porcentaje se realiza mediante una regla de 3 conociendo el número máximo de alumnos (chicos+chicas).

1.4.- Una tienda realiza durante un mes un descuento del 15% en todos sus productos, realizar un programa al cual se le introduce un precio y nos da por pantalla el precio con el 15% de descuento.

1.5.- Realizar un programa al que se le introduce una cantidad de kilobytes y este debe de convertirlos en megabytes. Se recomienda el uso de TRUNC.

1.6.- Realizar un programa que pida el sueldo de 4 empleados y a continuación les aplique un aumento del 10%, 12% y 15% respectivamente a los 4 sueldos.

1.7.- Realizar un programa que pida al usuario que introduzca una primera cadena de caracteres, a continuación se le pedirá una segunda cadena de caracteres para finalmente sacar por pantalla la concatenación de ambas cadenas de caracteres. Ejemplo:

Frase 1 -> Hola mi nombre es Antonio.

Frase 2-> Y soy vuestro profe de Programación.

Resultado -> Hola mi nombre es Antonio. Y soy vuestro profe de Programación.

1.8.- Realizar un programa que pida al usuario que introduzca un número real; a continuación sacar por pantalla la parte entera de dicho número. Ejemplo:

Numero real -> 67.9123

Parte Entera -> 67

1.9.- Realizar un programa que pida al usuario que introduzca una cadena de caracteres. A continuación se deberá de mostrar en pantalla la cadena de caracteres introducida por el usuario en mayúscula y en minúscula. Ejemplo:

Cadena -> HoLa Que tAl.

Resultado 1 -> HOLA QUE TAL.

Resultado 2-> hola que tal.

1.10.- Realizar un programa que pida la nota de un determinado alumno. A continuación redondear dicha nota al entero más próximo. Ejemplo:

Nota -> 7,75.

Resultado -> 8.

2. ALGORITMOS CON SECUENCIAS DE CONTROL.

2.1. IF. Completando el ejercicio 1.1.1 Realizar un algoritmo donde el usuario asigna los valores de las variables “Num1” y “Num2” y a continuación calcule su suma, su resta, su producto y su división, mostrando los resultados por pantalla. PERO, antes de realizar la división comprueba que Num2 no sea cero o no.

- Si Num2 es 0: Asigna el valor “Inf” a la variable de división.
- Si Num2 es mayor que 0: No realiza ninguna acción y el algoritmo funciona igual que el desarrollo en el ej. 1.1.1

2.2. IF. Realizar un algoritmo que solicite un número al usuario para las variable “Num1” y muestra por pantalla si es el número es par o impar.

- Si $\text{Num1} \text{ MOD } 2 == 0 \rightarrow \text{PAR}$
- Si $\text{Num1} \text{ MOD } 2 != 0 \rightarrow \text{IMPAR}$

2.3.1- IF. Realizar un algoritmo que solicite dos números al usuario para las variables “Num1” y “Num2” y a continuación muestre por pantalla uno de los siguientes mensajes:

- Si $\text{Num1} > \text{Num2}$: “El primer número ingresado es MAYOR que el segundo”.
- Si $\text{Num1} < \text{Num2}$: “El primer número ingresado es MENOR que el segundo”.

2.3.2.- IF. Completa el ejercicio anterior para un caso que no estaría contemplado anteriormente, mediante una nueva sentencia **IF** muestra por pantalla el texto “El primer número ingresado es IGUAL que el segundo” en caso de ser iguales ambos números.

2.3.3 - IF. Completando el ejercicio anterior se solicita que previo a la comparación de ambos números se valide que la entrada de datos es válida. Entonces:

- Si Num1 o $\text{Num2} < 0$. Las entradas son inválidas y finalizará el algoritmo con un mensaje de “Formato de entrada incorrecto” y no se continuará el algoritmo anterior.
- Si Num1 y $\text{Num2} \geq 0$. Entonces el formato es correcto y se ejecutará el resto de algoritmo desarrollado en los puntos 2.3.1 y 2.3.2.

2.4. IF. Realizar un algoritmo que solicite 4 números al usuario (Num1 , Num2 , Num3 y Num4) mediante sentencias IF calcula cual es el mayor y el menor número de la lista 4 números ingresados.

2.5.1 SWITCH. Realizar un algoritmo que solicite un número entre 1 y 7. Muestra por pantalla a qué día de la semana se refiere, siendo: 1=“Lunes”, 2=“Martes”, 3=“Miercoles”, 4=“Jueves”, 5=“Viernes”, 6=“Sábado”, 7=“Domingo”.

2.5.2. SWITCH. Realiza un algoritmo similar al anterior, pero en el caso de los meses del año solicite un número entre 1 y 12, siendo 1 “Enero” y 12 “Diciembre”. Se mostraría por pantalla el mes al que el usuario hace referencia.

2.6. FOR. Realizar un algoritmo que solicite un número. Mediante un bucle FOR muestra por pantalla la tabla de multiplicación (10 primeros números) del número ingresado.

2.7. WHILE. Realizar un algoritmo que solicite dos números al usuario y calcule tanto su máximo común divisor, el cuál mostrará por pantalla.

2.8.1. El segundero. Construye un segundero que muestre por pantalla las diferentes transiciones por lo que pasaría un minuterero con una espera entre cada mensaje de 1 segundo.

2.8.2 El minuterero & segundero. Completa el algoritmo anterior incorporando en esta ocasión también la transición de minutos y su correlación con la secuenciación generado por el segundero.

2.8.3 El minuterero & segundero - configurable. Terminado con nuestro algoritmo, al inicio del algoritmo solicitaremos el minuto y segundo de partida.

2.9.1. El perro dormilón. Tiene un perro muy dormilón que sin previo aviso se duerme, realiza un juego en el cuál debes adivinar si tu perro está durmiendo o despierto. Define una variable llamada "estado_perro" asignándole un valor aleatorio entre 0 y 1 "`<- azar(2)`". Construye un bucle WHILE donde solicite un dato (0 o 1) al usuario. El algoritmo:

1. Lee un dato por parte del usuario en la variable "lectura_usuario".
2. Debe validar que el dato de entrada está entre 0 y 1.
 - a. Si lectura_usuario=-1 finaliza el bucle WHILE y termina el programa.
 - b. Si lectura_usuario es distinto a [-1, 0, 1]. Enviar por pantalla el mensaje de: "Formato erróneo".
 - c. Si lectura_usuario es 0 o 1 prosigue con el algoritmo.
3. Re-calcula el valor de "estado_perro" utilizando la función "azar".
4. Compara el valor ingresado por el usuario y el nuevo valor de "estado_perro":
 - a. Si el estado_perro == lectura_usuario, saca por pantalla el mensaje "ACERTASTES".
 - b. Si el estado_perro != lectura_usuario, saca por pantalla el mensaje "FALLASTES".
5. El algoritmo se repetirá desde el punto 1 hasta que lectura_usuario sea -1

2.9.1. Continúa con el ejercicio 3. en este caso llega constancia de un contador de aciertos. Este contador volverá a 0 con cada fallo y sumará 1 a su valor actual con cada acierto seguido. Su valor se mostrará junto con cada mensaje de acierto o fallo con el siguiente formato:

- En caso de acierto, sacar por pantalla el mensaje "ACERTASTES. Van: X", siendo X el número del contador de aciertos en ese momento.
- En caso de fallo, sacar por pantalla el mensaje "FALLASTES. Tu racha de aciertos es: X", siendo X el número del contador de aciertos en ese momento.

3. ALGORITMOS CON PROCEDIMIENTOS Y FUNCIONES.

3.1. Similar al ejercicio 1.9. Realizar un programa que pida al usuario que introduzca una cadena de caracteres, en esta ocasión se debe realizar una llamada al procedimiento “realizarConversion” que realiza la acción requerida dentro de un nuevo subproceso, cuyo argumento de entrada es la propia sentencia introducida por el usuario.

3.2. Construye interfaz de usuario basada en comando (CLI) conforme dos ventanas, con una serie de opciones de navegación limitadas basada en números.

a. Ventana principal:

```
#####  
#####  
  
Boletin de ejercicios de programacion (CLI)  
  
Pantalla principal. Seleccione el bloque de ejercicios:  
  
1. Algoritmos con operadores numericos  
2. Algoritmos con secuencias de control  
-1. Exit  
  
#####  
#####
```

Las opciones 1 y 2 accederán a un listado diferente de la ventana secundaria.

La opción -1 sale del programa.

b. Ventana secundaria:

```
++++++  
  
Pantalla secundaria. Seleccione el ejercicio que desea:  
1. Ej1.1  
2. Ej1.2  
3. Ej1.3  
4. Ej1.4  
5. Ej1.5  
6. Ej1.6  
7. Ej1.7  
8. Ej1.8  
9. Ej1.9  
10. Ej1.10  
  
0. Atras  
-1. Exit  
  
++++++
```

Las opciones del 1 al 10 accederán mostrarán el siguiente texto: “Accediendo al ejercicio x.y” siendo x y los índices del ejercicio seleccionado.

> 2

Accediendo al ejercicio 1.2

+++++

Pantalla secundaria. Seleccione el ejercicio que desea:

1. Ej1.1
2. Ej1.2
3. Ej1.3
4. Ej1.4
5. Ej1.5
6. Ej1.6
7. Ej1.7
8. Ej1.8
9. Ej1.9
10. Ej1.10

0. Atras

-1. Exit

Luego, la opción 0 regresa a la ventana principal y la opción -1 termina el programa.