

Algoritmos y Programación I

Guía de ejercicios de estructuras de control: Secuenciales, Condicionales y repetitivas.

- 1.- En un estacionamiento cobran \$/. 1.500 por hora o fracción. Diseñe un algoritmo que determine cuanto debe pagar un cliente por el estacionamiento de su vehículo, conociendo el tiempo de estacionamiento en horas y minutos.
- 2.- Un supermercado ha puesto en oferta la venta al por mayor de cierto producto, ofreciendo un descuento del 15% por la compra de más de 3 docenas y 10% en caso contrario. Además por la compra de más de 3 docenas se obsequia una unidad del producto por cada docena en exceso sobre 3. Diseñe un algoritmo que determine el monto de la compra, el monto del descuento, el monto a pagar y el número de unidades de obsequio por la compra de cierta cantidad de docenas del producto.
- 3.- Una compañía dedicada al alquiler de automóviles cobra un monto fijo de \$300.000 para los primeros 300 km de recorrido. Para más de 300 km y hasta 1000 km, cobra un monto adicional de \$ 15.000 por cada kilómetro en exceso sobre 300. Para más de 1000 km cobra un monto adicional de \$ 10.000 por cada kilómetro en exceso sobre 1000. Los precios ya incluyen el 20% del impuesto general a las ventas, IVA. Diseñe un algoritmo que determine el monto a pagar por el alquiler de un vehículo y el monto incluido del impuesto.
- 4.- Diseñe un algoritmo que lea tres longitudes y determine si forman o no un triángulo. Si es un triángulo determine de qué tipo de triángulo se trata entre: equilátero (si tiene tres lados iguales), isósceles (si tiene dos lados iguales) o escaleno (si tiene tres lados desiguales). Considere que para formar un triángulo se requiere que: "el lado mayor sea menor que la suma de los otros dos lados".
- 5.- Realizar un algoritmo que pida un número y calcule su factorial (El factorial de un número es el producto de todos los enteros entre 1 y el propio número y se representa por el número seguido de un signo de exclamación. Por ejemplo $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$).

6.- Realizar un algoritmo que pida números hasta que se introduzca un cero. Debe imprimir la suma y la media de todos los números introducidos.

7.- Realizar un algoritmo que pida caracteres e imprima 'VOCAL' si son vocales y 'NO VOCAL' en caso contrario, el programa termina cuando se introduce un espacio.

8.- Realizar un algoritmo que imprima todos los números pares entre dos números que se le pidan al usuario.

9.- Realizar un algoritmo que muestre la tabla de multiplicar de un número introducido por teclado.

10.- Realizar un algoritmo que dados dos números, uno real (base) y un entero positivo (exponente), saque por pantalla el resultado de la potencia. No se puede utilizar el operador de potencia.

11.- Realizar un algoritmo que diga si un número introducido por teclado es o no primo. Un número primo es aquel que sólo es divisible entre él mismo y la unidad. Nota: Es suficiente probar hasta la raíz cuadrada del número para ver si es divisible por algún otro número.

12.- Mostrar en pantalla los N primeros números primos. Se pide por teclado la cantidad de números primos que queremos mostrar.

13.- Hacer un programa que lea una serie de números enteros positivos del teclado y calcule el valor máximo de los mismos y cuántas veces aparece dicho valor repetido. La entrada de datos finalizará cuando se introduzca un 0.

14.- Realizar un algoritmo que pida al usuario su nombre y contraseña y le de tres oportunidades para introducir los datos correctos, que serán root y 1234. Si los datos introducidos son correctos se mostrará por pantalla "Bienvenido al sistema". En caso

contrario se mostrará un mensaje por pantalla indicando que se ha superado el número de intentos permitido

15.- Dado un número entero determine la cantidad de dígitos que tiene.

16.- Dado un numero entero, calcular la cantidad de divisores que tiene e imprimirlos en pantalla.

17.- Dado un número entero determine la suma de sus dígitos.

18.- Realizar un algoritmo que lea dos números y presente una serie de opciones correspondientes a distintas operaciones que podemos realizar con ellos (suma, resta, multiplicación y división), de manera que, en función de la opción elegida, muestre el resultado de la operación realizada. En aquellos casos en los que se desee seguir operando con los mismos números, se debe contestar 'S' (Si a la siguiente pregunta "¿Otra operación con los mismos números (S/N)?" o 'N' (No), en caso contrario. Finalmente el programa deberá concluir tras responder 'S' (Si o 'N' (No) a la siguiente pregunta: "¿Terminar (S/N)?".

19.- Realice un algoritmo que permita leer un número positivo. Luego indique si es un número deficiente.

Nota: Un número deficiente es número natural que cumple que la suma de sus divisores propios es menor que el propio número. Por ejemplo, 16 es un número deficiente ya que sus divisores propios son 1, 2, 4 y 8 y se cumple que $1+2+4+8=15$, que es menor que 16.

20.- Se desea crear un programa para una playa de estacionamiento, que registre de cada auto que sale los 3 números de su patente, el modelo (año), la cantidad de horas de estacionamiento y el costo por hora. Imprimirá, para cada auto, su: N° de patente e Importe abonado. Al finalizar el día mostrará por pantalla: a) La cantidad de horas totales estacionadas b) ¿Cuántos autos de modelo posterior a 1995 estuvieron 2 horas o menos) ¿Cuál es la patente (3 últimos números) del auto que estuvo mayor cantidad de horas y cuántas horas estuvo?

21.- Una línea de autobuses cobra un mínimo de 20 euros por persona y trayecto. Si el trayecto es mayor de 200 km el billete tiene un recargo de 3 céntimos por km adicional. Sin embargo,

para trayectos de más de 400 km el billete tiene un descuento del 15 %. Por otro lado, para grupos de 3 o más personas el billete tiene un descuento del 10 %. Con las consideraciones anteriores, escriba un algoritmo que lea por teclado la distancia del viaje a realizar, así como el número de personas que viajan juntas. Con ello se debe calcular tanto el precio del billete individual como el total a pagar si viaja más de una persona.

22.- Usando bucles como únicas estructuras de control y una única sentencia de impresión para el carácter “o”, escriba un algoritmo que imprima la siguiente salida, una pirámide invertida de dicho caracter:

```
ooooo
  ooo
    o
```

23.- Haga un algoritmo que dados dos números introducidos por teclado indique si son amigos o no. El algoritmo debe ofrecer la posibilidad de continuar. Nota: Dos números amigos son dos enteros positivos (n_1 , n_2) tales que la suma de los divisores propios de uno de ellos es igual al otro (la unidad se considera divisor propio, pero no lo es el mismo número). Por ejemplo, 220 y 284 son amigos, ya que: Suma de divisores de 284: $1 + 2 + 4 + 71 + 142 = 220$ Suma de divisores de 220: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$. También son números amigos 17296 y 18416

24.- Simular la división usando solamente restas. Dados dos números enteros mayores de uno, desarrollar una clase ejecutable en Java que calcule el cociente y el residuo usando sólo restas. Método: Restar el dividendo del divisor hasta obtener un resultado menor que el divisor, este resultado es el residuo, y el número de restas realizadas es el cociente. Por ejemplo: $50 / 13$: $50 - 13 = 37$ una resta realizada $37 - 13 = 24$ dos restas realizadas $24 - 13 = 11$ tres restas realizadas 11 es menor que 13, entonces: el residuo es 11 y el cociente es 3

25.- Desarrolle algoritmo que convierta un número entero en otro número entero que será el primero pero con las cifras que lo forman escritas al revés. Ejemplo: convertirá el número entero 1842 en el 2481.

26.- Mostrar la siguiente secuencia de números: 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9

27.- Leer 20 números y mostrar:

- Cuántos son positivos y cuánto suman
- Cuántos son negativos y cuánto suman
- Cuántos son ceros

28.- Diseña un algoritmo que permita leer dos números enteros positivos y luego determine el máximo común divisor y el mínimo común múltiplo de dichos números.

29.- Realice un algoritmo que permita leer un número positivo. Luego indique si es un número perfecto.

Nota: Un número es perfecto si la suma de sus divisores (sin incluir al mismo número) resulta igual a sí mismo. Ejemplo: 6 es perfecto debido a que $6 = 1 + 2 + 3$

30.- Realizar un algoritmo que permita cargar dos listas de 5 valores cada una. Informar con un mensaje cuál de las dos listas tiene un valor acumulado mayor (mensajes "Lista 1 mayor", "Lista 2 mayor", "Listas iguales").

31.- Realizar un algoritmo llamado "Cuenta_uno", que reciba un número n e imprima la cantidad de "unos" (1) que se encuentran en los dígitos de n.

32.- Crear un programa que muestre en consola el siguiente patrón.

```
*
**
***
****
*****
*****
*****
*****
```

```
*****
*****
*****
*****
*****
*****
```

33.- Elabora un programa que simule un reloj que inicia en 0 horas 0 minutos 0 segundos y termine en 23 horas 59 minutos y 59 segundos.

