**ESTRUCTURAS DE REPETICIÓN**

# *1. Ejercicio*

Construir un programa en el que se ingresen números enteros hasta que se hayan ingresado 5 números pares e informar si alguno de ellos es también múltiplo de cuatro.

# *2. Ejercicio*

Realizar un programa que reciba una cantidad indefinida de números enteros positivos hasta que se ingresa el 0. A continuación el programa debe indicar cuál fue el mayor, cuál el menor y el promedio general de valores ingresados (no incluir el 0 de finalización). Considerar la posibilidad de que no se ingresen números (lo primero que se ingresa es 0).

# 3. Ejercicio

Hacer un programa que permita ingresar una cantidad indefinida de números reales. La carga finaliza cuando se ingresa 0 (cero). A continuación, debe informarse cuántos números ingresados fueron números enteros.

# 

# *4. Ejercicio*

Escribir un programa que reciba una cantidad indefinida de números enteros positivos hasta que se ingresa el 0. A continuación el programa debe indicar cuáles fueron los 2 mayores. Considerar la posibilidad de que no se ingresen números (lo primero que se ingresa es 0) o se ingresen menos de 2 números.

# 5. Ejercicio

Elaborar un programa que solicite al usuario que ingrese un número entero (de cualquier cantidad de dígitos) y determine si el mismo es un número capicúa.

# *6. Ejercicio*

Para encontrar el máximo común divisor (MCD) de dos números se emplea el algoritmo de Euclides, que puede describirse así: Dados dos enteros ***a*** y ***b***, se divide ***a*** por ***b***. Si el resto de la división es cero, el número ***b*** es el máximo común divisor. Si la división no es exacta, el valor de ***a*** pasa a ser ***b*** y el nuevo valor de ***b*** será el resto de la división anterior. Se repite el procedimiento hasta obtener un resto cero. El último divisor (***b***) es el MCD buscado. Escribir un programa que calcule el MCD de dos números enteros. Por ejemplo, si ***a*** = 2366 y ***b***=273, se divide ***a*** por ***b*** y el resto es 182; como no es cero, se divide el nuevo ***a*** que será 273 por el nuevo ***b*** que será 182 y da como resto 91; como no es cero se divide 182 por 91 y da como resto cero; se concluye entonces que 91 es el MCD entre 2366 y 273.

# *7. Ejercicio*

Hacer un programa que encuentre e imprima un listado con todos los números de 4 cifras que cumplan la condición de que la suma de los dígitos de orden impar es igual a la suma de los dígitos de orden par. Por ejemplo, el número 7821 cumple esta condición ya que 7 + 2 = 8 + 1.

# *8. Ejercicio*

Armar un programa en el que se ingrese un número entero decimal (base diez) y convertirlo a binario.

# *9. Ejercicio*

Confeccionar un programa en el cual se ingrese una cantidad indefinida de números enteros positivos, hasta ingresar 0 (cero). Finalizada la carga, el programa debe informar si los números ingresados (excluido el 0) se ingresaron en orden ascendente y, si éste fuera el caso, también indicar si fueron números consecutivos o no. Por ejemplo, si se ingresa 13, 14 y 15 se indicará que fueron ingresados en orden ascendente y consecutivo. Si se ingresa 13, 14, 16 y 17 se indicará que fueron ingresados en orden ascendente y no consecutivo. Considerar la posibilidad de que no se ingresen números (lo primero que se ingresa es 0).

# *10. Ejercicio*

Escribir un programa que permita ingresar las notas de una cantidad indefinida de alumnos. A continuación el programa deberá mostrar la cantidad de alumnos aplazados (nota menor a 4), la cantidad de alumnos aprobados (nota entre 4 y 7 inclusive) y la cantidad de alumnos que promocionan la materia (nota superior a 7). En cada caso, se mostrará el porcentaje del total de notas cargadas que cada caso representa y el promedio general de todas las notas. Las notas pueden ser valores reales y la carga finaliza cuando la nota ingresada es 0. Validar que las notas estén dentro del rango de 1 a 10, imprimiendo un mensaje de error por cada valor no válido y descartándolo. Ejemplo:

Ingrese nota: ***5***

Ingrese nota: ***4***

Ingrese nota: ***81***

Error. Nota no válida. Valor ignorado.

Ingrese nota: ***2***

...

Ingrese nota: 0

Cantidad de aplazos: 5 (10%)

Cantidad de aprobados: 15 (30%)

Cantidad de promocionados: 30 (60%)

Promedio general: 6.72

# 

# 

# ***11. Ejercicio***

Elaborar un programa que solicite al usuario que ingrese un número entero y determine si el mismo es un número primo.

# *12. Ejercicio*

Escribir un programa que encuentre los primeros 4 números perfectos. Un número perfecto es un entero positivo, que es igual a la suma de todos los enteros positivos (excluido él mismo) que son divisores del número. Por ejemplo, el primer número perfecto es 6, ya que los divisores de 6 son 1, 2, 3 y 1 + 2 + 3 = 6.

# 13. Ejercicio

Elaborar un programa que calcule el factorial de un número entero. El programa principal debe solicitar el ingreso de un número entero positivo (validarlo), calcular y luego mostrar por pantalla su factorial. Ejemplo:

Ingrese un número entero: ***5***

El factorial de 5 es: 120

# 14. Ejercicio

Confeccionar un programa para calcular utilidades de inversión en plazos fijos. Para esto, se ingresarán las cantidades que serán los importes mensuales que el cliente desea invertir (cantidad = 0 para terminar). Luego se debe obtener el resultado de la colocación total en plazos fijos.

Primero se debe solicitar la Tasa Nominal Anual (TNA) al principio de la serie de importes, a partir de la cual se obtiene la Tasa Efectiva Mensual (TEM = TNA /12). En cada mes, se calculan los intereses multiplicando la TEM por el total de montos acumulados hasta dicho mes.

Al finalizar la carga, se imprimirá el monto final (suma de montos invertidos) más los intereses acumulados durante cada mes. Ejemplo: Se desea invertir tres sumas (durante tres meses seguidos) en plazos fijos.

Ingrese porcentaje de TNA: ***8***

Ingrese monto del mes 1: ***2000***

Ingrese monto del mes 2: ***3000***

Ingrese monto del mes 3: ***1000***

Ingrese monto del mes 4: ***0***

El monto final es: 6086.67

Ayuda:

* Ej.: TNA = 8% = 0.08. TEM = TNA /12 = 0.08/12 = 0.0067
* Intereses = 2000 \* 0.0067 + (2000+3000) \* 0.0067 + (2000+3000+1000) \* 0.0067
* Monto total = Inversión + Intereses = 2000 + 3000 + 1000 + 86.67 = 6086.67

# 15. Ejercicio

Realizar un programa en el que se ingresa un texto de longitud indefinida (hasta ingresar ENTER). Luego debe informarse por pantalla cuántas de sus letras fueron mayúsculas y cuántas fueron minúsculas. Por ejemplo, para el texto “Hola Mundo” debe obtenerse 2 mayúsculas y 7 minúsculas

# 

# 16. Ejercicio

Confeccionar un programa en el que se ingresa una letra a buscar y luego un texto de tamaño indefinido (hasta ingresar ENTER). Luego, el programa debe informar cuántas apariciones de la letra hay en ese texto. Ejemplo:

Ingrese la letra a contar: ***a***

Ingrese el texto: ***La bella y graciosa moza, marchose a lavar la ropa.***

La letra "a" aparece 11 veces.

# 17. Ejercicio

Realizar un programa en el que se ingresa un texto de longitud indefinida (hasta ingresar ENTER). Luego informar cuántas de las letras ingresadas fueron vocales (contar tanto mayúsculas como minúsculas) y cuántas fueron consonantes (contar tanto mayúsculas como minúsculas). Por ejemplo, para el texto “Kate Austen fue una de las 6 sobrevivientes del Oceanic 815” debe informarse que hay 22 vocales y 23 consonantes.

# 18. Ejercicio

Confeccionar un programa que permita el ingreso de letras en minúscula y que brinde al usuario tres operaciones posibles a través de un menú de opciones. La primera opción devolverá la mayúscula correspondiente. La segunda opción devolverá la letra siguiente en el alfabeto (si la letra ingresada fuera “***z***” se debe informar que es la última). La tercera opción devolverá la letra anterior en el alfabeto siempre que la letra ingresada no sea “***a***”. Validar que efectivamente el carácter ingresado sea una letra minúscula o ***\**** (este último determina el final de la serie de opciones). Ejemplo:

Ingrese letra: ***f***

1 – Convertir a mayúscula

2 – Obtener la siguiente

3 – Obtener la anterior

Escoja una operación (1 – 3): 1

La mayúscula es F.

# 19. Ejercicio

Efectuar un programa que solicite al usuario que ingrese la base de un triángulo rectángulo. Luego dibujar en la pantalla dicho triángulo como se muestra en el siguiente ejemplo:

Ingrese base: ***4***

\*

\*\*

\*\*\*

\*\*\*\*

# 

# 20. Ejercicio

Efectuar un programa que solicite al usuario que ingrese un número entero impar (validar el valor ingresado). El programa tendrá que dibujar en la pantalla un triángulo de asteriscos cuya base sea el valor ingresado. Ejemplo:

Ingrese número: ***8***

Número no válido. Ingrese número: 11

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

# 21. Ejercicio

Efectuar un programa que solicite al usuario que ingrese la diagonal de un rombo (validar que el valor ingresado sea impar). Luego deberá dibujar dicho rombo en la pantalla. Ejemplo:

Ingrese diagonal: ***8***

Valor incorrecto. Ingrese diagonal: ***7***

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

# 

# 22. Ejercicio

Efectuar un programa que solicite al usuario que ingrese la base y la altura de un rectángulo. Luego deberá dibujar en la pantalla el rectángulo hueco. Ejemplo:

Ingrese base: ***10***

Ingrese altura: ***4***

\*\*\*\*\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*\*\*\*\*

# 

# 23. Ejercicio

Hacer un programa por cada dibujo que se ejemplifica tal que se solicite al usuario el ingreso del tamaño de la figura (validar que sea impar). Luego deberá mostrar por pantalla el dibujo en cuestión. Los dibujos que cada programa debe hacer deben respetar las proporciones del ejemplo.

| |  |  | | --- | --- | |  |  | | |  |  | | --- | --- | |  |  | | |  |  | | --- | --- | |  |  | | |  |  | | --- | --- | |  |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

# 24. Ejercicio

En una pared de un cierto tamaño debe colocarse una guarda. La guarda viene en piezas de 2 tamaños diferentes y cada pieza no puede fraccionarse. Hacer un programa donde se ingrese la siguiente información:

* La longitud de la pared (en centímetros)
* La longitud de las piezas grandes (en cm).
* La cantidad de piezas grandes
* La longitud de las piezas chicas (en cm)
* La cantidad de piezas chicas

A continuación, el programa deberá mostrar todas las combinaciones de piezas que cubran toda la longitud de la pared, de forma exacta (sin sobrantes ni faltantes). Alternativamente, si no es posible completar la pared, imprimir el mensaje “La pared no puede cubrirse con esas guardas”. Ejemplo:

Ingrese longitud de la pared: ***47***

Ingrese cantidad de piezas chicas: ***20***

Ingrese longitud de piezas chicas: ***3***

Ingrese cantidad de piezas grandes: ***21***

Ingrese longitud de piezas grandes: ***8***

1 piezas de 8 cm + 13 piezas de 3 cm

4 piezas de 8 cm + 5 piezas de 3 cm