

Programación

Ejercicios Tema 4 Estructuras de Control Repetitivas

Autores:

M. Paz Sesmero Lorente
Paula de Toledo Heras
Fco. Javier Ordoñez Morales
Juan Gómez Romero
Jose A. Iglesias Martínez
Jose Luis Mira

1. Escriba un programa en C que muestre en pantalla los números enteros del 1 al 100.
2. Escriba un programa en C que muestre en pantalla los números enteros del 1 al 100 de 2 en 2.
3. Escriba un programa en C que muestre en pantalla los números enteros del 100 al 1.
4. Escriba un programa que muestre en pantalla todos los múltiplos de 3 entre los números 1 y 100.
5. Escriba un programa en C que genere la siguiente serie: 5,10,15,20,25,30... El programa preguntará primero cuantos términos se quieren mostrar y después mostrará la serie correspondiente.
6. Escriba un programa que pida al usuario un número, y muestre el mensaje "has introducido el numero ---" (mostrando el número que ha sido introducido). Esta acción debe repetirse hasta que el usuario introduzca el número 0. En ese momento se mostrará el mensaje "Finalizando: Se ha introducido el número 0". Además, se debe mostrar cuántos números se han introducido y su suma.
7. Escriba un programa 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.

Notas:

Puesto que las cadenas de caracteres finalizan con el carácter nulo (`'\0'`), usuario y contraseña se declararán como cadenas de longitud 5 (`char usuario[5]`). Para leer cualquiera de estas cadenas con la función `scanf` se usará el descriptor de formato `%s` y dado que el nombre de la cadena ya es una dirección, éste no debe ir precedido del símbolo `&` (`scanf("%s", usuario)`).

Para comparar cadenas se debe usar la función **strcmp** (*string compare*), por ejemplo `strcmp(usuario, "root")`, que devolverá un 0 si las cadenas son iguales y otro valor si son distintas.

8. Escriba un programa que eleve un número a una potencia, utilizando dos métodos: a) utilizando la función potencia y b) usando un bucle, sin recurrir al operador potencia. Nota: En C, para poder usar la función potencia (`pow(x, y)=xy`) es necesario incluir la biblioteca **math** (`#include <math.h>`).

9. Escriba un programa en C que solicite 10 números enteros positivos al usuario y calcule y muestre los siguientes valores:
 - La suma de todos los números leídos.
 - La media de los números.
 - El mayor número introducido.
 - El menor número introducido.
10. Escriba un programa en C que solicite números al usuario hasta que se hayan introducido 10 números o la suma de todos los números leídos sea mayor que 100. A continuación mostrar un mensaje indicando qué condición se ha cumplido (es decir, si se han introducido 10 números o si su suma es mayor que 100).
11. Escribir un programa en C para adivinar un número entre 1 y 100 que previamente se ha definido como una constante. El programa irá pidiendo números al usuario y, siempre que dicho número no coincida con el número secreto, le indicará si el número introducido es mayor o menor que el número secreto que tiene que adivinar. Al final, el programa indicará la cantidad de intentos que se han necesitado para adivinar el número. Si el número de intentos es menor que 5 se mostrará “Enhorabuena!”. Si es un valor entre 5 y 10 se mostrará el mensaje “No está mal”. Si el número de intentos es mayor que 10 se mostrará el mensaje “Debe practicar más”.
12. Modificar el ejercicio anterior para que el número máximo de intentos sea 5. Al terminar el programa indicará si el usuario ha ganado (si ha adivinado el número en menos de cinco intentos) o no.
13. Escriba un programa en C que solicite al usuario números positivos de tres o más cifras y compruebe si son pares. Si el número introducido es positivo pero de una o dos cifras, se solicitará un nuevo número. El programa finaliza cuando se introduce “0” o un número negativo.
14. Escriba un programa en C que permita calcular el importe de una factura, a partir de una serie de artículos vendidos. Para ello, el programa irá preguntando para cada artículo la cantidad de unidades vendidas y el precio unitario.
El programa comprobará que tanto el precio como la cantidad son números positivos, y en caso contrario volverá a solicitar los valores.
La lectura de artículos acabará cuando se introduzca un 0 en la primera pregunta. Entonces se imprimirá por pantalla el importe total de la factura.