

INTRODUCCIÓN A LA PROGRAMACIÓN I

Ejercicios Propuestos Tema 4: Estructuras de Selección y Control

Contenido

1. Ejercicio 1. Raíces reales de una ecuación de grado 2	1
2. Ejercicio 2. Máximo Común Divisor	1
3. Ejercicio 3. Máximo, Mínimo y Media.....	1
4. Ejercicio 4. Número Primo.....	1
5. Ejercicio 5. Número Perfecto	1
6. Ejercicio 6. Función seno $(2x) - x$	1
7. Ejercicio 7. Raíz cuadrada.....	2
8. Ejercicio 8. Multiplicación escalar y vectorial	2
9. Ejercicio 9. Media de los naturales de un rango	2
10. Ejercicio 10. Representación binaria de un numero de dos cifras	2
11. Ejercicio 11. Suma Serie aritmética	3
12. Ejercicio 12. Suma de números introducidos por el usuario	3
13. Ejercicio 13. Cociente de dos números	3
14. Ejercicio 14. Calculadora	3
15. Ejercicio 15. Año bisiesto.....	3
16. Ejercicio 16. Números primos	3
17. Ejercicio 17. Producto como sumas	4
18. Ejercicio 18. Días del mes	4
19. Ejercicio 19. Tres números ordenados	4
20. Ejercicio 20. Tablas de multiplicar	4

1. Ejercicio 1. Raíces reales de una ecuación de grado 2

Calcular las raíces reales de un polinomio de grado 2, expresado por el producto de tres términos cada uno integrado por un el producto de un coeficiente y las potencias de x elevadas a 2, 1 y 0.

$$a x^2 + b x + c = 0$$

El programa pedirá al usuario los valores de “a”, “b” y “c”, para luego calcular las raíces del polinomio. Hay que asegurar que los valores introducidos son correctos. Usar la función `sqrt` de la librería `math.h` cuyo prototipo es:

`double sqrt(double x);` Returns the square root of $x^2/3$

2. Ejercicio 2. Máximo Común Divisor

Escribir un programa que pida dos números enteros y positivos al usuario, y que calcule el máximo común divisor (MCD). Utilizar el algoritmo de Euclides

- Se divide el número mayor entre el menor.
- Si la división es exacta, el divisor es el m.c.d.
- Si la división no es exacta, dividimos el divisor entre el resto obtenido y continuamos de esta forma hasta obtener una división exacta.
- El m.c.d. es el último divisor.

3. Ejercicio 3. Máximo, Mínimo y Media

Escribir un programa que calcule y visualice el más grande, el más pequeño y la media de n números ($n > 0$). El valor de n se solicitará al principio del programa y los números serán introducidos por el usuario.

4. Ejercicio 4. Número Primo

Comprobar si un número entero es primo, teniendo en cuenta que un primo solo tiene 2 divisores

5. Ejercicio 5. Número Perfecto

Un número perfecto es un entero, que es igual a la suma de todos los enteros positivos (excluido el mismo) que son divisores del número. El primer número perfecto es el 6 ($1+2+3=6$)

- Escribir un programa que pida un número entero positivo y determine si es perfecto
- Escribir un programa que determine los 6 primeros números perfectos

6. Ejercicio 6. Función seno (2x) - x

Escribir un programa que presente en modo tabla los valores de la función $f(x)=\text{seno}(2x)-x$ para los valores $x = 0, 0.5, 1, 1.5, 2 \dots 9, 9.5, 10$. Utilizar la función `sin` de `<math.h>`

7. Ejercicio 7. Raíz cuadrada

El algoritmo de raíz cuadrada por el método babilónico se basa en la idea de que el lado de un cuadrado es a su vez la raíz cuadrada del área. Por lo tanto, la idea es buscar dos números muy similares que, multiplicados entre sí, den el número del que se busca su raíz. Se pide crear un programa que realice el cálculo de la raíz cuadrada usando el método babilónico (paso 2 del programa). El programa funcionará de la siguiente manera:

1. Pedir un número al usuario, y un margen de error
2. Realizar el cálculo de la raíz cuadrada
 - 2.1.1 Escoja dos números b y h tales que $bh=x$, siendo x el número del que hay que calcular la raíz
 - 2.1.2 Si $h \simeq b$ (h es aproximadamente igual a b o lo que es equivalente que su diferencia sea menor que el margen de error), vaya al paso 2.6, si no, vaya al paso 2.3
 - 2.1.3 Asigne $b \leftarrow \frac{h+b}{2}$
 - 2.1.4 Asigne $h \leftarrow \frac{x}{b}$
 - 2.1.5 Vaya al paso 2.2
 - 2.1.6 El resultado es b
3. Mostrar el resultado por pantalla, y el error obtenido en el cálculo

8. Ejercicio 8. Multiplicación escalar y vectorial

Crear un programa que pida dos vectores de 3 dimensiones (tres números por vector, en total, seis números), y que calcule la multiplicación escalar y vectorial entre ellos. El programa funcionará de la siguiente manera:

- Pedir 3 números para el primer vector
- Pedir 3 números para el segundo vector
- Elegir operación: 1- Multiplicación escalar, 2- Multiplicación vectorial
- Mostrar resultado

9. Ejercicio 9. Media de los naturales de un rango

Crear un programa que calcule la media de la suma de los números naturales dentro de un rango dado por el usuario. El programa pedirá dos números enteros, sumará los valores intermedios (incluyendo los dos números dados) y mostrará la media. Además, deberá asegurarse que los datos introducidos por el usuario concuerden, en caso contrario dará un error.

Ej: Para el intervalo de números 7..24, la media de la suma de sus números sería 15,5 (279/18)

10. Ejercicio 10. Representación binaria de un numero de dos cifras

Crear un programa que pida por pantalla un número de dos cifras, y que muestre su representación binaria, sin ceros a la izquierda. Se usarán los operadores de desplazamiento (<<,>>), máscaras de bits, operaciones binarias (&, |,etc..) que se consideren necesarias.

Ejemplo: Para el número 16, el programa mostrará 10000

11. Ejercicio 11. Suma Serie aritmética

Realizar la suma de una serie aritmética de 1 a n, siendo n un dato introducido por el usuario, imprimiendo al final el valor de la suma y el de la formula $n*(n+1)/2$. Comprueba que el resultado es el mismo

Utilizar un bucle for

12. Ejercicio 12. Suma de números introducidos por el usuario

Hacer un programa que calcule la suma de los números introducidos por un usuario hasta que éste introduzca un cero

13. Ejercicio 13. Cociente de dos números

Hacer un programa que calcula el cociente de dos números controlando que no se pueda dividir por cero. Si el denominador es cero terminar el programa

14. Ejercicio 14. Calculadora

Escribir un programa que realice una calculadora. Las operaciones que se pueden realizar deben mostrarse en bucle para que el usuario decida. Para elegir cada una de ellas es necesario introducir la inicial de la operación que se desea realizar y se admite que sea en mayúsculas o minúsculas.

```
MENÚ PRINCIPAL
[S/s] - Sumar
[R/r] - Restar
[M/m] - Multiplicar
[D/d] - Dividir
[E/e] - Exit
```

La idea es que este menú se repita todo el rato hasta que el usuario seleccione la opción 'E' o 'e'. En caso de introducir una opción inválida, habrá que mostrar un mensaje de error y volver al menú. Mediante un switch habrá que distinguir qué opción ha sido elegida y mostrar por pantalla el resultado de la operación elegida. No se puede llamar en ningún momento a la función exit().

15. Ejercicio 15. Año bisiesto

Hacer un programa que calcule si un año es bisiesto. Se pedirá al usuario por pantalla el año del que se quiere calcular si es o no bisiesto. Un año es bisiesto si es múltiplo de 4. Además, los años múltiplos de 100 sólo son bisiestos cuando a la vez son múltiplos de 400 (por ejemplo 1800 no es bisiesto mientras que 2000 si lo es). Hay que asegurar que el año que se introduce está comprendido entre el año 1 y el 2021

16. Ejercicio 16. Números primos

Hacer un programa que calcula los números primos contenidos entre 1 y 100 sabiendo que un número es primo si sólo es divisible entre 1 y entre sí mismo.

17. Ejercicio 17. Producto como sumas

Hacer un programa que calcule el producto de dos números naturales utilizando solo sumas

18. Ejercicio 18. Días del mes

Escribir un programa que devuelva el número de días que tiene un mes. A continuación, se muestran los detalles que se deben tener en cuenta:

- El mes es introducido por el usuario por teclado (número entre 1 y 12).
- Se considera que el año no es bisiesto.
- Se debe comprobar que el número está dentro del rango y si no fuera así se le debe volver a pedir al usuario.

19. Ejercicio 19. Tres números ordenados

Codificar un programa que lea tres números enteros introducidos uno detrás del otro por teclado y comprobar que los números siguen una secuencia de orden ascendente. Escribir un mensaje que así lo muestre. Comprobar que se están introduciendo números enteros y si no es así se debe volver a pedir al usuario.

20. Ejercicio 20. Tablas de multiplicar

Codificar un programa que presente las tablas de multiplicar del 1 al 10.

21. Ejercicio 21. Inversión de cifras

Codificar un programa que invierta los dígitos de un número entero positivo. A continuación, se muestran los detalles que debe cumplir el programa:

- El número es introducido por el usuario por teclado.
- Se debe comprobar que el usuario ha introducido un número entero positivo.

22. Ejercicio 21. Contar dígitos de un número entero

Codificar un programa que cuente y muestre la cantidad de dígitos en un número entero ingresado por el usuario