

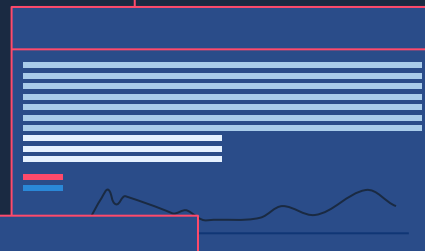


Tecnicatura en Programación Universitaria

Comisión

B

Clase 04



Clase Practica

Martes 30 de Abril 2024

Ing. Oviedo Codigoni Carlos Nicolas



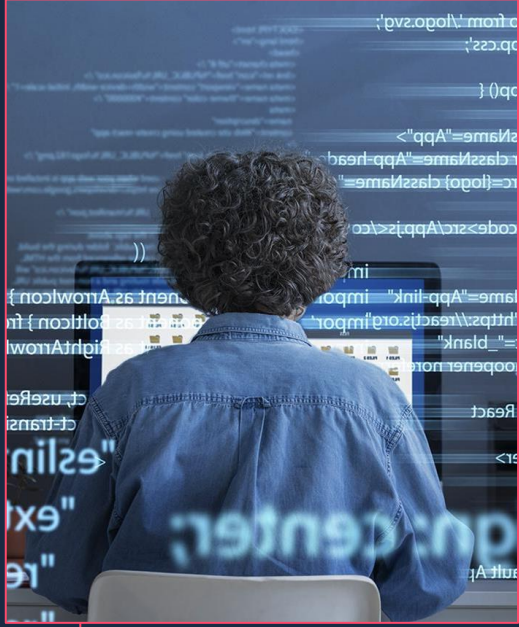
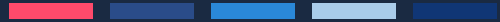
Norma ISO/IEC 9899:1999



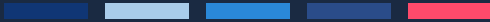
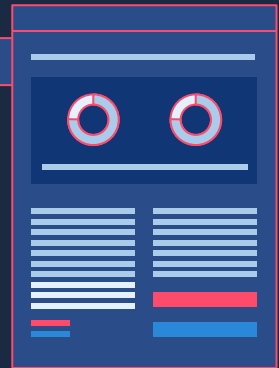
La norma C99, o ISO/IEC 9899:1999, es una revisión importante del estándar del lenguaje de programación C. Fue publicada por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC) en el año 1999. Esta norma fue desarrollada por el Comité Técnico Conjunto ISO/IEC JTC1/SC22/WG14, que se encarga de la estandarización del lenguaje de programación C. La C99 introdujo nuevas características de lenguaje, bibliotecas estándar adicionales y mejoras en la portabilidad, lo que la convierte en una revisión significativa del estándar C.

- | | | |
|---------------------------|---------------------------|-----------------------------|
| 1. <code>stdio.h</code> | 9. <code>fenv.h</code> | 17. <code>stdarg.h</code> |
| 2. <code>stdlib.h</code> | 10. <code>float.h</code> | 18. <code>time.h</code> |
| 3. <code>stdint.h</code> | 11. <code>iso646.h</code> | 19. <code>complex.h</code> |
| 4. <code>stddef.h</code> | 12. <code>limits.h</code> | 20. <code>inttypes.h</code> |
| 5. <code>stdbool.h</code> | 13. <code>locale.h</code> | 21. <code>tgmath.h</code> |
| 6. <code>assert.h</code> | 14. <code>math.h</code> | 22. <code>wchar.h</code> |
| 7. <code>ctype.h</code> | 15. <code>setjmp.h</code> | |
| 8. <code>errno.h</code> | 16. <code>signal.h</code> | |





...<Math.h>...



#include <math.h>

La librería math en C proporciona un conjunto de funciones matemáticas para operaciones comunes y avanzadas. Estas funciones pueden ser utilizadas para realizar cálculos trigonométricos, exponenciales, logarítmicos, redondeo y otras operaciones matemáticas.

1 - Funciones Trigonómicas

- sin ()
- cos ()
- tan ()
- asin ()
- acos ()
- atan ()
- sinh ()
- cosh ()
- tanh ()
- atan2 ()

2 - Funciones Exponenciales y Logarítmicas

- exp()
- log()
- log10()
- pow()
- sqrt()

3 - Funciones de Redondeo y Valor Absoluto

- ceil()
- floor()
- fabs()

4 - Otras Funciones Matemáticas

- Fmod
- remainder()
- hypot()
- round()

include<exercise.h>

1. **Cálculo del valor absoluto de un número:** Solicita al usuario ingresar un número y muestra su valor absoluto utilizando la función `fabs()`.
2. **Comprobación de número par o impar:** Pide al usuario ingresar un número y utiliza una estructura `if` para determinar si es par o impar. Puedes utilizar la función `fmod()` de `math.h` para determinar si el número es divisible por 2.
3. **Calculadora de área:** Solicita al usuario que elija una figura geométrica (cuadrado, círculo, triángulo) y calcule su área. Utiliza una estructura `switch` para seleccionar la figura y las funciones correspondientes de `math.h` para realizar los cálculos.
4. **Cálculo de raíces de una ecuación cuadrática:** Solicita al usuario los coeficientes de una ecuación cuadrática y calcula sus raíces. Utiliza la fórmula cuadrática y las funciones `sqrt()` y `pow()` de `math.h` dentro de una estructura `if` para manejar diferentes casos (raíces reales, raíces imaginarias).
5. **Clasificación de años bisiestos:** Pide al usuario ingresar un año y determina si es bisiesto o no. Utiliza una estructura `if/else` para aplicar las reglas de los años bisiestos.
6. **Cálculo de descuento de compra:** Pide al usuario ingresar el precio de un producto y su cantidad. Calcula el monto total de la compra y aplica un descuento del 10% si la cantidad es mayor o igual a 10. Utiliza una estructura `if/else` para aplicar el descuento.

include<exercise.h>

1. **Verificación de números capicúa con bucle for:** Pide al usuario ingresar un número entero positivo y utiliza un bucle for para verificar si es un número capicúa o no.
2. **Cálculo de la raíz cuadrada de un número con bucle while:** Pide al usuario ingresar un número positivo y utiliza un bucle while para calcular su raíz cuadrada.
3. **Calculadora de números perfectos con bucle do-while:** Pide al usuario ingresar un número entero positivo y utiliza un bucle do-while para comprobar si es un número perfecto o no.
4. **Generador de secuencia de números triangulares con bucle while:** Pide al usuario ingresar un número entero positivo N y utiliza un bucle while para imprimir los primeros N números triangulares.
5. **Verificación de números primos con bucle for:** Pide al usuario ingresar un número entero y utiliza un bucle for para verificar si es primo o no.
6. **Calculadora de números armónicos:** Pide al usuario ingresar un número entero positivo N y calcula la suma de los primeros N términos de la serie armónica utilizando un bucle for.

Conceptos Matematicos

- **Números Perfectos:** Un número perfecto es un número entero positivo que es igual a la suma de sus divisores propios positivos, excluyendo al número mismo.

Por ejemplo:

1. el primer número perfecto es 6, ya que sus divisores propios son 1, 2 y 3, y $1 + 2 + 3 = 6$.
2. Otro ejemplo es 28, cuyos divisores propios son 1, 2, 4, 7 y 14, y $1 + 2 + 4 + 7 + 14 = 28$.

- **Números Triangulares:** Un número triangular es un número que se puede representar como la suma de los primeros n números naturales consecutivos para algún valor de n .

Por ejemplo, 1, 3, 6, 10, 15, etc., son números triangulares.

El nombre proviene de la disposición de puntos en una forma triangular. El n -ésimo número triangular se puede calcular usando la fórmula: $T_n = \frac{n*(n+1)}{2}$, donde T_n es el número triangular y n es el número natural.

Ejemplo:

$$n=3$$

$$T_3 = \frac{3 * (3 + 1)}{2} = \frac{3 * 4}{2} = \frac{12}{2} = 6$$

Por lo tanto, el tercer número triangular es 6.

- **Números armónicos:** son una serie matemática que surge en el estudio de la teoría de números y en la física, especialmente en el análisis de circuitos eléctricos y en la acústica. La serie armónica es una serie infinita de la forma:

$H_n = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$, Donde $H(n)$ representa el n -ésimo número armónico y n es un número natural mayor que cero.