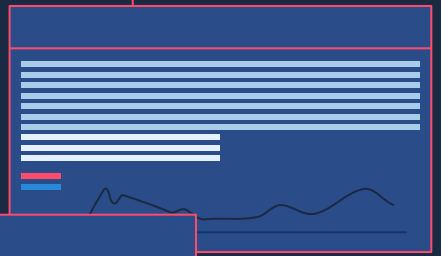
Tecnicatura en Programación Universitaria

Comisión

B

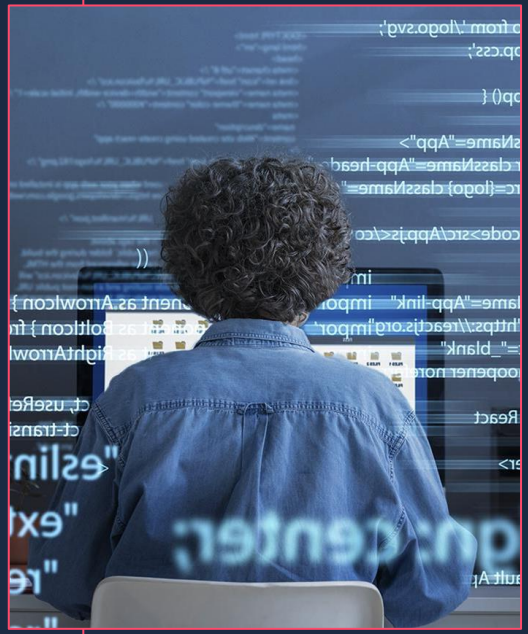
Clase 05



Clase Practica

Martes 07 de Mayo 2024

Ing. Oviedo Codigoni Carlos Nicolas



/*

< Punteros & Procedimientos >

*/

PUNTEROS

Un puntero es una variable que almacena una dirección de memoria y por ello, se suele decir que se “apunta” a dicha posición de memoria.

VARIABLES

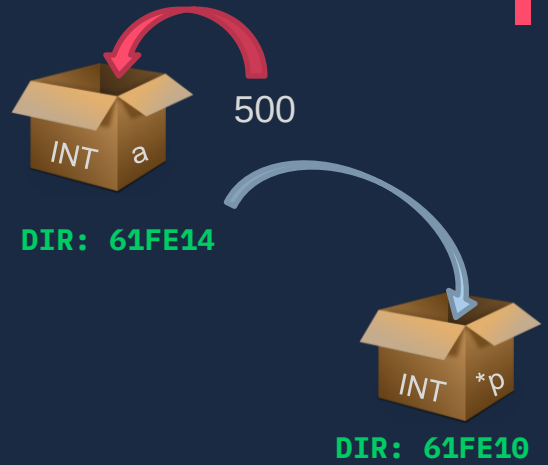
- **Nombre**: relacionado
- **Tipo**:
  - *int*
  - *unsigned int*
  - *short*
  - *unsigned short*
  - *long*
  - *unsigned long*
  - *long long*
  - *unsigned long long.*
  - *float*
  - *double*
  - *long double*
  - *Char*
  - *signed char*
  - *unsigned char*
- **Valor:** Dato

PUNTERO

- **Nombre:** relacionado
- **Tipo**:
  - *int*
  - *unsigned int*
  - *short*
  - *unsigned short*
  - *long*
  - *unsigned long*
  - *long long*
  - *unsigned long long.*
  - *float*
  - *double*
  - *long double*
  - *Char*
  - *signed char*
  - *unsigned char*
- **Valor**: Dirección de Memoria

¡¡¡ IMPORTANTE !!!

- **Operador de indirección [*]**: *Acceder al valor apuntado por un puntero.*
- **Operador de dirección [&]**: *Obtener la dirección de memoria de una variable.*



PROCEDIMIENTOS & FUNCIONES

- **Procedimiento:** *es un conjunto de instrucciones con una tarea bien definida y un nombre (identificador) que se utiliza para "invocar" (ejecutar) este bloque de instrucciones. La invocación puede ser hecha el número de veces que sea necesario dentro del programa, incluso dentro del mismo procedimiento.*
- **Función**: *es un tipo especial de procedimiento que entrega (retorna) un valor como resultado.*

TIPOS DE PROCEDIMIENTOS

- *sin retorno (void).*
- *con retorno.*
- *con parámetros.*
- *sin parámetros.*

SINTAXIS PROCEDIMIENTOS

- *Declaración*
- *Definición.*
- *Llamado.*

PASAJE DE PARAMETROS

- *Paso de parámetros por valor.*
- *Paso de parámetros por referencia (utilizando punteros).*

```c
#include <stdio.h>

// Declaración de la función
int suma(int a, int b);

int main() {
    int resultado;
    // Llamada a la función
    resultado = suma(5, 3);
    printf("La suma es: %d\n", resultado);
    return 0;
}

// Definición de la función
int suma(int a, int b) {
    return a + b;
}
```

```c
int suma(int a, int b);
```
Declaración

```c
resultado = suma(5, 3);
```
Llamado

```c
int suma(int a, int b)
{
    return a + b;
}
```
Definición

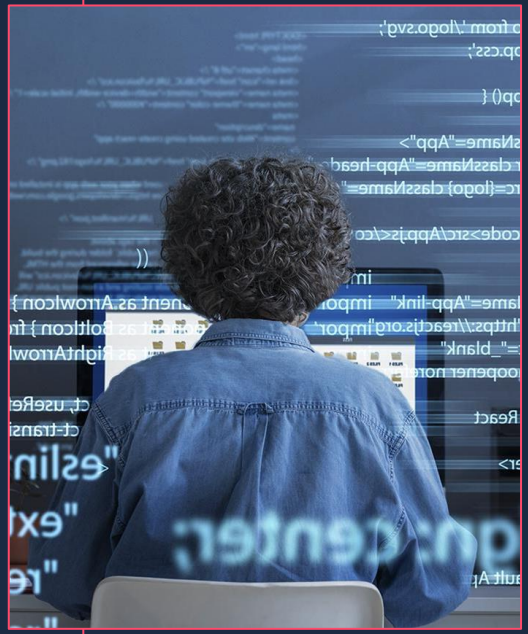
include<exercise.h>

PUNTEROS

1. **Intercambio de Valores:** *Realice un programa que tome dos variables enteras como argumentos y las intercambie utilizando punteros. Mostrar las direcciones de las variables y del puntero.*
2. **Duplicación de un Número:** *Realice un programa que tome un puntero a un número entero y duplique su valor (es decir, multiplique el valor por 2). Mostrar la dirección de la variable y del puntero.*
3. **Suma y Resta:** *Realice un programa que tome dos variables enteras y dos punteros a enteros. La función debe calcular la suma y la resta de las dos variables enteras y almacenar los resultados en los punteros proporcionados. Mostrar las direcciones de las variables y de los punteros.*
4. **Incremento y Decremento:** *Realice un programa que tome un puntero a un número entero y realice un incremento y un decremento en su valor. Mostrar la dirección de la variable y del puntero.*
5. **Cambio de Signo:** *Realice un programa que tome un puntero a un número entero y cambie su signo (por ejemplo, si el número es positivo, debe convertirse en negativo y viceversa). Mostrar la dirección de la variable y del puntero.*

PROCEDIMIENTOS

A. **Calcular el Máximo:** *Escribe una función en C llamada* **maximo** *que tome dos números enteros como parámetros y devuelva el mayor de los dos.*
B. **Verificar Número Primo:** *Escribe una función en C llamada* **es-primo** *que tome un número entero como parámetro y devuelva 1 si es primo y 0 si no lo es.*
C. **Calcular Potencia:** *Escribe una función en C llamada potencia que tome dos números enteros como parámetros y devuelva el resultado de elevar el primero a la potencia del segundo.*
D. **Verificar Par o Impar:** *Escribe una función en C llamada* **es-par** *que tome un puntero a un número entero como parámetro y devuelva 1 si es par y 0 si es impar.*
E. **Incremento y Decremento:** *Escribe una función en C llamada* **incrementar-decrementar** *que tome un puntero a un número entero como parámetro y realice un incremento y un decremento en su valor. El incremento de cremento dependerá del parámetro opción (0 , 1).*



/*

< Vectores & Matrices >

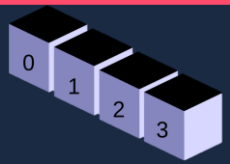
*/

Vectores & Matrices

En términos de programación en C, tanto los vectores como las matrices se declaran y manipulan de manera similar, pero difieren en su dimensionalidad y cómo se accede a sus elementos.

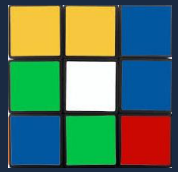
VECTORES

- Un vector, también conocido como array en C, es una colección contigua de elementos del mismo tipo almacenados en memoria.
- Cada elemento en un vector tiene un índice único que se utiliza para acceder a él.
- Los vectores en C pueden ser unidimensionales (un array) o multidimensionales (arrays de arrays), lo que significa que pueden tener una o más dimensiones.
- Los vectores son útiles cuando necesitamos almacenar y trabajar con una lista ordenada de elementos del mismo tipo, como números, caracteres o estructuras de datos más complejas.



MATRICES

- Una matriz es una estructura de datos bidimensional que organiza los elementos en filas y columnas.
- En C, las matrices se pueden representar como arrays multidimensionales, donde cada elemento es accesible mediante dos índices: uno para la fila y otro para la columna.
- Las matrices son útiles para representar datos tabulares, imágenes, gráficos y para resolver problemas que involucran relaciones bidimensionales entre los elementos.

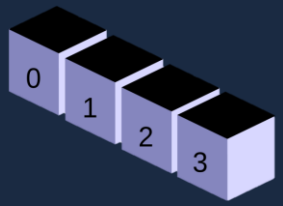


Vectores & Matrices y sus Partes

```c
#include <stdio.h>

int main()
{
    // Declaración e inicialización de un vector de enteros
    int vector[5] = {2, 4, 6, 8, 10};

    // Accediendo e imprimiendo los elementos del vector
    printf("Elementos del vector:\n");
    for (int i = 0; i < 5; i++) {
        printf("vector[%d] = %d\n", i, vector[i]);
    }
}
```




```c
int vector [5] = {2, 4, 6, 8, 10};
```

Tipo

Dimensión

Valor

Nombre

Posiciones

```c
#include <stdio.h>

int main()
{
    // Declaración e inicialización de una matriz de enteros 3x3
    int matriz[3][3] = {
                        {1, 2, 3},
                        {4, 5, 6},
                        {7, 8, 9}
                        };
    // Accediendo e imprimiendo los elementos de la matriz
    printf("Elementos de la matriz:\n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("matriz[%d][%d] = %d\n", i, j, matriz[i][j]);
        }
    }
}
```


```c
int matriz [3][3] = {
    {1, 2, 3},
    {4, 5 , 6},
    {7, 8 , 9}
};
```

Tipo

Dimensiones

Valor

Nombre

Filas

Columnas

#include<exercise.h>

1. **Suma de vectores:** *Escribe un programa que solicite al usuario ingresar dos vectores de igual longitud y luego calcule la suma de estos vectores.*

2. **Producto escalar de vectores:** *Crea un programa que tome dos vectores de la misma longitud como entrada y calcule su producto escalar.*

3. **Promedio de elementos:** *Escribe un programa que solicite al usuario ingresar un vector de números enteros y luego calcule y muestre el promedio de los elementos en el vector.*

4. **Transposición de una matriz:** *Escribe un programa que tome una matriz como entrada y calcule su matriz transpuesta. La matriz transpuesta se obtiene intercambiando filas por columnas.*

5. **Multiplicación de matrices:** *Implementa un programa que tome dos matrices como entrada y calcule su producto matricial. Asegúrate de que las dimensiones de las matrices sean compatibles para la multiplicación.*

6. **Determinante de una matriz cuadrada:** *Crea un programa que calcule el determinante de una matriz cuadrada ingresada por el usuario. Puedes usar cualquier método de cálculo de determinante que prefieras (por ejemplo, el método de Laplace o la regla de Sarrus para matrices 3x3).*

Conceptos Matemáticos

***La matriz traspuesta**, es una operación matricial que consiste en intercambiar las filas por las columnas de una matriz dada. En otras palabras, si tienes una matriz A, su matriz traspuesta, denotada como A^T, se obtiene cambiando cada elemento a_{ij} de A por el elemento a_{ji} en la posición opuesta en A^T.*

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

***El producto matricial** es una operación fundamental en álgebra lineal que se realiza entre dos matrices y produce otra matriz como resultado. La regla básica para el producto matricial es que el número de columnas de la primera matriz debe ser igual al número de filas de la segunda matriz.*

$$C_{ij} = \sum_{k=1}^{n} a_{ik} \times b_{kj}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \; x \; B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = C = \begin{bmatrix} a_{11} \times b_{11} + a_{12} \times b_{21} & a_{11} \times b_{12} + a_{12} \times b_{22} \\ a_{21} \times b_{11} + a_{22} \times b_{21} & a_{21} \times b_{12} + a_{22} \times b_{22} \end{bmatrix}$$

Conceptos Matemáticos

Una matriz cuadrada *es una matriz que tiene el mismo número de filas que de columnas. Es decir, una matriz* $n \times n$ *donde* n *es un número entero positivo.*

Matriz 2x2 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Matriz 3x3 $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

El determinante de una matriz cuadrada es un número escalar que se calcula a partir de los elementos de la matriz y que proporciona información importante sobre la matriz en cuestión. Específicamente, para una matriz cuadrada A de tamaño $n \times n$, su determinante se denota como $|A|$ o $det(A)$.

Matriz 2x2 $$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

Matriz 3x3 $$\det\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\right) = a(ei - fh) - b(di - fg) + c(dh - eg)$$