|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Marco Antonio Martinez Quintana |
| *Asignatura:* | Estructura de Datos y Algoritmos I |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 2 |
| *Integrante(s):* | Enzo Valdés Zavala |
| *No. de Equipo de cómputo empleado:* |  |
| *No. de Lista o Brigada:* |  |
| *Semestre:* | 2020-1 |
| *Fecha de entrega:* | 9 de febrero del 2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivos:**

Utilizar apuntadores en lenguaje C para acceder a las localidades de memoria tanto de datos primitivos como de arreglos.

**Introducción:**

Un apuntador es una variable que contiene una dirección de memoria a otra variable.

Esta hace referencia a la localidad de memoria de otra variable.  
Se utilizan para dar claridad y simplicidad a las operaciones a nivel de memoria.  
Los apuntadores tienen funciones de bajo nivel, ya que permiten manipular directamente la memoria RAM de la computadora.  
  
Para declarar un apuntador:  
**TipoDeDato \*apuntador, variable;**

Para asignarle un valor al apuntador:

**apuntador = &variable;**Dentro de las funciones, se pueden trabajar con apuntadores.Estos se pueden trabajar con valores o con las referencias de las variables.  
Cuando se trabaja con los valores se envía una copia del valor original a la función, de manera que si se modifica el contenido de la variable, la original permanece.  
Mientras que si se trabaja con referencias, se envía un apuntador hacia el valor original y se termina trabajando con dicho valor todo el tiempo.  
  
Otro uso de los apuntadores es como recorren arreglos:

* apArr++: Incrementa apArr para apuntar a la siguiente localidad de memoria.
* apArr+=i: Incrementa apArr para apuntar a la i-ésima localidad de memoria a partir del valor inicial de apArr.

**Desarrollo:**

**Explicación:**

Durante esta practica, ejecutamos diferentes codigos que nos permitían la visualización de las diferentes funciones de un apuntador:

1-. Apuntando a un carácter

2-. Apuntando a un entero, asignándole valores

3-. Apuntando a un arreglo

4-. Paso de variables por valor y por referencia

5-. Aritmetica de direcciones

6-. Arreglo undimensional

7-. Arreglo bidimensional

8-. Y el cifrado cesar

1-. A screenshot of a social media post

Description automatically generated

A screenshot of a computer

Description automatically generated

2-.A screenshot of a social media post

Description automatically generatedA screenshot of a computer

Description automatically generated

3-.A screenshot of a social media post

Description automatically generatedA screenshot of a computer screen

Description automatically generated

4-. A screenshot of a social media post

Description automatically generatedA screenshot of a computer screen

Description automatically generated

5-.

A screenshot of a social media post

Description automatically generatedA screenshot of a computer

Description automatically generated

6-.A screenshot of a social media post

Description automatically generatedA screenshot of a computer screen

Description automatically generated

7-.A screenshot of a social media post

Description automatically generatedA screenshot of a computer

Description automatically generated

8-.

A screenshot of a computer

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a computer screen

Description automatically generated

**Conclusión:**

Aprendimos el uso adecuado de los apuntadores, sus diferentes usos y aplicaciones, y un poco de criptografía romana.

**Bibliografía:**

-El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

-Teoria vista en el salón de clases.