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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* |  |
| *Asignatura:* |  |
| *Grupo:* |  |
| *No de Práctica(s):* |  |
| *Integrante(s):* |  |
|  |  |
|  |  |
| *Semestre:* |  |
| *Fecha de entrega:* | 5 de marzo de 2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

### Práctica no.2: Aplicaciones de apuntadores

**Objetivo:**

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

|  |
| --- |
| ***Ejercicio 1***  Escribir un programa que cree una matriz de caracteres de 16x16 , iniciar la matriz con números del 0 al 255, después imprime  la matriz utilizando solo un apuntador y un solo ciclo de repetición, en el siguiente formato :  **<dirección de memoria>  : <valor numérico> <carácter>**  Ejemplo:  0061FED0 : 40  (  0061FED1 :  41  )  0061FED2 : 42  \*  0061FED3 : 43  +  Nota: Prueba utilizar unsigned char.   1. ¿Qué representan estos datos? 2. Imprime en pantalla el tamaño de la matriz por medio de la función sizeof(). 3. Si la matriz fuera de tipo **int**, que es lo que cambiara en la impresión.   [3 puntos] |
| [Remplazar por el código fuente] |

|  |
| --- |
| Ejercicio 2 Escribir un programa que hago lo siguiente:   1. Cree tres arreglos de apuntadores a entero de nombre datos, pares e impares, los tres para 4 elementos. 2. Leer cuatro números y hacer la relación con cada elemento del arreglo datos. 3. Relacionar los valores par e impar con el arreglo del mismo nombre. 4. Imprimir los valores pares utilizando notación indexada. 5. Imprimir los valores impares utilizando aritmética de apuntador. 6. Hacer un esquema de la memoria al finalizar el programa.   [3 puntos] |
| [Remplazar por el código fuente] |

|  |
| --- |
| Ejercicio 3 Analizar el programa practica02\_e03.c y responder lo siguiente:   1. Indicar que ocurre con la función imprimeArreglo( ) que debería mostrar en pantalla el arreglo. 2. ¿En qué tipo de parámetro se convierte el arreglo para enviarlo a la función ? 3. ¿Qué se debe cambiar en el programa?   [2 puntos] |
| [Remplazar por el código fuente] |

|  |
| --- |
| Ejercicio 4 El objetivo del ejercicio es utilizar los argumentos de la función main para leer valores desde la línea de comando.  Analizar el programa practica02\_e04.c y responder lo siguiente:   1. ¿Qué tipo de variable es: char \*argv[] ?   Respuesta:\_\_\_\_\_\_   1. Imprimir el valor de argc, ¿Qué representa este dato?   Respuesta:\_\_\_\_\_\_   1. Imprimir el número de Bytes de la variable argv, utilizando sizeof(). ¿Por qué dice 4?   Respuesta:\_\_\_\_\_\_   1. Imprimir el contiene cada elemento de argv[], ¿Qué representa ?   Respuesta:\_\_\_\_\_\_\_  Ejemplo:  argv[0] <DIRECCION>  argv[1] <DIRECCION>  ...   1. Colocar un apuntador en la dirección de cada valor de argv[], e imprimir letra por letra hasta encontrar el final.     ap <- argv[0]    <DIRECCION ap>+0 : 'h'  <DIRECCION ap>+1 : 'o'  <DIRECCION ap>+2 : 'l'  <DIRECCION ap>+3 : 'a'  ap <- argv[1]    <DIRECCION ap>+0 : 'a'  ...    [2 puntos] |
| [Remplazar por el código fuente] |

## Conclusión