



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): M.I. Heriberto García Ledezma

Asignatura: Estructura de datos y algoritmos I

Grupo: 15

No de Práctica(s): practica 2

Integrante(s): Fuentes Llantada Marco Antronio

Rojas Contreras Aaron

No. de lista o brigada: 11 y 31

Semestre: 2025-2

Fecha de entrega: 28/02/2025

Observaciones:

CALIFICACIÓN: _____

Objetivos de la práctica

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

Ejercicios de la práctica

Ejercicio 1. Programa que pregunte al usuario su edad, su estatura en metros y su letra favorita. Después que imprima en pantalla la dirección de memoria dónde se almacena cada uno de los datos que proporcionó. Por último, que imprima en pantalla cada uno de los datos, pero utilizando apuntadores a cada variable que los almacena.

```
#include <stdio.h>

int main() {
    int edad;
    float estatura;
    char letra;

    int *P_edad = &edad;
    float *P_estatura = &estatura;
    char *P_letra = &letra;

    printf("ingrese su edad: ");
    scanf("%d", &edad);

    printf("ingrese su estatura: ");
    scanf("%f", &estatura);

    while (getchar() != '\n'); //profe esto fue porque saltaba lo de letra

    printf("ingrese su letra favorita: ");
    scanf("%c", &letra);

    printf("\nDirecciones de memoria:\n");
    printf("edad: %p\n", (void*)P_edad);
    printf("estatura: %p\n", (void*)P_estatura);
    printf("letra: %p\n", (void*)P_letra);

    printf("\nValores utilizando apuntadores:\n");
    printf("edad: %d\n", *P_edad);
    printf("estatura: %2f\n", *P_estatura);
    printf("letra: %c\n", *P_letra);

    return 0;
}
```

```
rexta@DESKTOP-J5AEL8D ~/EDA
$ ./a.exe
ingrese su edad: 21
ingrese su estatura: 1.74
ingrese su letra favorita: M

Direcciones de memoria:
edad: 0x7fffffcc24
estatura: 0x7fffffcc20
letra: 0x7ffffcc1f

Valores utilizando apuntadores:
edad: 21
estatura: 1.740000
letra: M
```

problema 2.

Apuntadores y cadenas de caracteres

2. Programa que pregunte al usuario su nombre completo con apellidos. Después que imprima en pantalla la dirección de memoria dónde se almacenó.

Por último, que imprima en pantalla el nombre que escribió el usuario primero utilizando el identificador del arreglo y luego un apuntador a la variable dónde se almacenó el nombre.

```
#include <stdio.h>

int main() {
    char nombre[101];
    char *P_nombre = nombre;

    printf("ingrese su nombre completo: ");
    fgets(nombre, sizeof(nombre), stdin);

    printf("\nDireccion de memoria donde se almaceno el nombre: %p\n", (void*)P_
    printf("nombre usando el apuntador: %s", P_nombre);

    return 0;
}
```

```
rexta@DESKTOP-J5AEL8D ~/EDA
$ ./a.exe
ingrese su nombre completo: Marco Antonio Fuentes Llantada

Direccion de memoria donde se almaceno el nombre: 0x7ffffcbdb0
nombre usando el apuntador: Marco Antonio Fuentes Llantada
```

problema 3. Aritmética de caracteres Programa que declare un arreglo de tres caracteres y lo inicialice con los valores a, b y c. Después, que haga lo siguiente:

A) mediante aritmética de direcciones de memoria imprima en pantalla los valores del arreglo separados por espacio. Escriba un printf por cada elemento del arreglo.

B) mediante aritmética de direcciones de memoria imprima en pantalla los valores del arreglo separados por espacio. Utilice un ciclo para automatizar la impresión de elementos de arreglo y no tener que escribir varios printf.

```
#include <stdio.h>

int main() {

    char arreglo[] = {'a', 'b', 'c'};
    char *P = arreglo;

    printf("A) %c %c %c\n", *P, *(P + 1), *(P + 2));

    printf("B) ");
    for (int i = 0; i < 3; i++) {
        printf("%c ", *(P + i));
    }
    printf("\n");

    return 0;
}
```

```
rexta@DESKTOP-J5AEL8D ~/EDA
$ ./a.exe
A) a b c
B) a b c
```

Actividad 4. Paso de argumentos por valor. Programa que pida dos números reales al usuario y que desde la función principal invoque a una función llamada Suma. La función suma recibirá los valores a sumar, los sumará y regresará la función que la invocó el resultado. Desde la función principal se imprimirá el resultado de la suma. Nota: No use variables globales.

```
#include <stdio.h>

float Suma(float a, float b) {
    return a + b;
}

int main() {
    float num1, num2, resultado;

    printf("Ingrese el primer número: ");
    scanf("%f", &num1);
    printf("Ingrese el segundo número: ");
    scanf("%f", &num2);

    resultado = Suma(num1, num2);

    printf("La suma de %.2f y %.2f es: %.2f\n", num1, num2, resultado);

    return 0;
}
```

```
rexta@DESKTOP-J5AEL8D ~/EDA
$ ./a.exe
Ingrese el primer número: 12
Ingrese el segundo número: 34
La suma de 12.00 y 34.00 es: 46.00
```

Programa 5. Programa que pida dos números reales al usuario y que desde la función principal invoque a una función llamada SumaMejorada a la que le pasara direcciones de memoria de los operandos y el resultado...

<pre>1 #include <stdio.h> 2 3 void SumaMejorada(float *num1, float *num2, float *resultado); 4 5 int main() { 6 float numero1, numero2, resultado; 7 8 printf("Ingrese el primer número real: "); 9 scanf("%f", &numero1); 10 11 printf("Ingrese el segundo número real: "); 12 scanf("%f", &numero2); 13 14 SumaMejorada(&numero1, &numero2, &resultado); 15 16 printf("El resultado de la suma es: %.2f\n", resultado); 17 18 return 0; 19 } 20 21 void SumaMejorada(float *num1, float *num2, float *resultado) { 22 *resultado = *num1 + *num2; 23 }</pre>	<pre>redmute@red:~/Documents/EDA1/Prácticas/practica2\$./programa5 Ingrese el primer número real: 3.1416 Ingrese el segundo número real: 2.42 El resultado de la suma es: 5.56</pre>
--	---

Programa 6 Programa que desde la función principal pregunte al usuario 5 números reales y los almacene en un arreglo. Después invocará a una función secundaria a la que le pase el arreglo...

```
1  #include <stdio.h>
2
3  float funcionSecundaria(float arreglo[], int tamano);
4
5  int main() {
6      float numeros[5];
7      float suma;
8      int i;
9
10     for (i = 0; i < 5; i++) {
11         printf("Ingrese el número real %d: ", i + 1);
12         scanf("%f", &numeros[i]);
13     }
14
15     suma = funcionSecundaria(numeros, 5);
16
17     printf("\nElementos del arreglo después de incrementar:\n");
18     for (i = 0; i < 5; i++) {
19         printf("Elemento %d: %.2f\n", i + 1, numeros[i]);
20     }
21
22     printf("\nLa suma de los elementos incrementados es: %.2f\n", suma);
23
24     return 0;
25 }
26
27 float funcionSecundaria(float arreglo[], int tamano) {
28     float suma = 0;
29     int i;
30
31     for (i = 0; i < tamano; i++) {
32         arreglo[i] += 10;
33         suma += arreglo[i];
34     }
35
36     return suma;
37 }
```

```
Ingrese el número real 1: 3.96
Ingrese el número real 2: 4.98
Ingrese el número real 3: 5.34
Ingrese el número real 4: 6.12
Ingrese el número real 5: 7.22
```

```
Elementos del arreglo después de incrementar:
Elemento 1: 13.96
Elemento 2: 14.98
Elemento 3: 15.34
Elemento 4: 16.12
Elemento 5: 17.22
```

```
La suma de los elementos incrementados es: 77.62
```

Programa 7 Programa que desde la función principal solicite al usuario una frase célebre escrita en minúsculas. El programa guardará la frase. Desde la función principal invocará a una función auxiliar que recibirá el arreglo que almacena la frase...

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void convertirAMayusculas(char frase[]);
5
6  int main() {
7      char frase[100];
8
9      printf("Ingrese una frase célebre en minúsculas: ");
10     fgets(frase, sizeof(frase), stdin);
11
12     convertirAMayusculas(frase);
13
14     printf("La frase en mayúsculas es: %s", frase);
15
16     return 0;
17 }
18
19 void convertirAMayusculas(char frase[]) {
20     int i = 0;
21
22     while(frase[i] != '\0') {
23         if(frase[i] >= 'a' && frase[i] <= 'z') {
24             frase[i] = frase[i] - 32;
25         }
26         i++;
27     }
28 }
```

```
Ingrese una frase célebre en minúsculas: facultad de ingeniería
La frase en mayúsculas es: FACILTAD DE INGENIERÍA
```

Programa 8 Programa que desde la función principal solicite al usuario una frase célebre escrita en minúsculas. El programa guardará la frase...

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void convertirAMayusculas(char *frase);
5
6  int main() {
7      char frase[100];
8
9      printf("Ingrese una frase célebre en minúsculas: ");
10     fgets(frase, sizeof(frase), stdin);
11
12     convertirAMayusculas(frase);
13
14     printf("La frase en mayúsculas es: %s", frase);
15
16     return 0;
17 }
18
19 void convertirAMayusculas(char *frase) {
20     while(*frase != '\0') {
21         if(*frase >= 'a' && *frase <= 'z') {
22             *frase = *frase - 32;
23         }
24         frase++;
25     }
26 }
```

```
redmute@red:~/Documents/EDA1/Prácticas/practica2$ ./programa8
Ingrese una frase célebre en minúsculas: facultad de ingeniería
La frase en mayúsculas es: FACULTAD DE INGENIERÍA
```

Conclusiones

Utilizar apuntadores en el lenguaje C permite acceder y manipular directamente las localidades de memoria de datos primitivos y arreglos, lo que proporciona una gestión eficiente de los recursos y una mayor flexibilidad en la programación. Los apuntadores permiten modificar valores sin necesidad de utilizar variables globales, optimizar el uso de la memoria y mejorar el rendimiento de los programas al evitar copias innecesarias de datos. Además, facilitan el acceso dinámico a estructuras de datos complejas, como listas enlazadas y matrices multidimensionales. Sin embargo, su uso incorrecto puede derivar en errores difíciles de depurar, como accesos indebidos a memoria o fugas de memoria, por lo que es fundamental manejarlos con precaución y una comprensión clara de su funcionamiento.