

Ejercicios Punteros C

Bruno Vega 5º 1º Electronica

Ejercicio N°1

1. Escribe un programa que declare una variable entera y un puntero a entero. Asigna un valor a la variable y luego asigna la dirección de memoria de esa variable al puntero. Imprime en pantalla el valor de la variable y el valor al que apunta el puntero.

```
#include <stdio.h>

int main()
{
    int var;
    int *Pvar;
    var = 7;
    Pvar = &var;

    printf("El valor de la variable es %d\n", var);
    printf("El valor al que apunta el puntero es %d\n", *Pvar);
    return 0;
}
```

Ejercicio N°2

2. Escribe un programa que declare dos variables racionales, asigna valores a cada una y luego intercambia los valores utilizando punteros. Imprime en pantalla los valores intercambiados.

```
#include <stdio.h>

int main()
{
    float a = 1.4, b = 2.6;
    printf("Numeros antes de intercambiar: Num1 = %.2f  Num2 = %.2f\n", a, b);
    float *Pa = &a, *Pb = &b;
    float temp = *Pa;
    *Pa = *Pb;
    *Pb = temp;
    printf("Numeros despues de intercambiar: Num1 = %.2f  Num2 = %.2f\n", a, b);
    return 0;
}
```

Ejercicio N°3

3. Escribe un programa que declare una variable entera, asigna un valor a la variable y luego duplica el valor utilizando un puntero. Imprime en pantalla el valor duplicado.

```
#include <stdio.h>

int main()
{
    int num = 4;
    int *Pnum = &num;
    printf("Numero original: %d\n", num);
    *Pnum = *Pnum * 2;
    printf("Numero duplicado: %d\n", num);
    return 0;
}
```

Ejercicio N°4

4. Escribe un programa, sin usar funciones, que declare un arreglo de enteros y un puntero a entero. Asigna la dirección de memoria del primer elemento del arreglo al puntero. Imprime en pantalla los valores del arreglo accediendo a través del puntero.

```
#include <stdio.h>

int main()
{
    int arr[] = {1,2,3,4,5,6,7,8,9,10};
    int *Parr = &arr;
    for(int i = 0; i < 10; i++){
        printf("%d\n", *Parr);
        ++ Parr;
    }
    return 0;
}
```

Ejercicio N°5

5. Escribe un programa, sin usar funciones, que declare un arreglo de enteros y calcule la suma de todos los elementos utilizando aritmética de punteros.

```
#include <stdio.h>

int main()
{
    int arr[] = {1,2,3,4,5,6,7,8,9,10};
    int *Parr = &arr;
    int suma = 0;
    for(int i = 0; i < 10; i++){
        suma += *Parr;
    }
}
```

```
        ++ Parr;  
    }  
    printf("La suma de los numeros del array es: %d\n", suma);  
    return 0;  
}
```

Ejercicio N°6

6. Escribe un programa, sin usar funciones, que reciba una cadena de caracteres del usuario y determine su longitud utilizando punteros.

```
#include <stdio.h>  
  
int main()  
{  
    char string[] = "Este es mi string";  
    int count = 0;  
    for (int i = 0; string[i] != '\0'; i++)  
        count++;  
    printf("El string tiene %d caracteres\n", count);  
    return 0;  
}
```

Ejercicio N°7

7. Escribe una función llamada "swap" que reciba dos punteros a enteros y los intercambie entre sí. En el programa principal, declara dos variables enteras, asigna valores a cada una y llama a la función "swap" pasando los punteros como argumentos. Imprime en pantalla los valores intercambiados.

```
#include <stdio.h>  
  
void swap(int* Ptr1, int* Ptr2);  
  
int main()  
{  
    int a = 33, b = 66;  
    int *Pa = &a, *Pb = &b;  
    swap(Pa, Pb);  
    printf("Valores Intercambiados: %d %d\n", *Pa, *Pb);  
    return 0;  
}  
  
void swap(int* Ptr1, int* Ptr2){  
    float temp = *Ptr1;  
    *Ptr1 = *Ptr2;  
    *Ptr2 = temp;  
}
```

Ejercicio N°8

8. Escribe una función llamada "duplicarNumero" que reciba un puntero a un número entero y duplique su valor. En el programa principal, declara una variable entera, asigna un valor a la variable, llama a la función "duplicarNumero" pasando el puntero de la variable como argumento, y muestra en pantalla el valor duplicado.

```
#include <stdio.h>

void duplicarNumero(int *Pnum);

int main()
{
    int numero = 7;
    int *Ptrnumero = &numero;
    duplicarNumero(Ptrnumero);
    printf("Numero duplicado: %d\n", numero);
    return 0;
}

void duplicarNumero(int *Pnum){
    *Pnum = *Pnum * 2;
}
```

Ejercicio N°9

9. Escribe una función llamada "inicializarArray" que reciba un puntero a array, el tamaño del array y el valor con el que se quiere inicializar. La función debe inicializar un array con el valor entero dado por argumento de la función. El array debe ser asignado a través del puntero. Muestra en pantalla el array inicializado.

```
#include <stdio.h>

void inicializarArray(int *ptr, int size, int valor);

int main() {
    int array[5];
    int valorInicial = 10;
    inicializarArray(array, 5, valorInicial);
    printf("Array inicializado:\n");
    for (int i = 0; i < 5; i++)
        printf("%d ", array[i]);
    printf("\n");
    return 0;
}

void inicializarArray(int *ptr, int size, int valor) {
    for (int i = 0; i < size; i++) {
        *(ptr + i) = valor;
    }
}
```

```
}
```

Ejercicio N°10

10. Escribe una función llamada "contarVocales" que reciba un puntero a una cadena de caracteres y cuente la cantidad de vocales que contiene. La función debe retornar el número total de vocales encontradas. En el programa principal, declara una cadena de caracteres, llama a la función "contarVocales" pasando el puntero a la cadena y muestra en pantalla el número de vocales encontradas.

```
#include <stdio.h>

int contarVocales(char *Pstr);

int main() {
    char cadena[] = "Este es un string de prueba"; //Contine 9 vocales
    printf("Número de vocales encontradas: %d\n", contarVocales(cadena));
    return 0;
}

int contarVocales(char *Pstr) {
    int count = 0;
    while (*Pstr != '\0') {
        if (*Pstr == 'a' || *Pstr == 'e' || *Pstr == 'i' || *Pstr == 'o' ||
            *Pstr == 'u' || *Pstr == 'A' || *Pstr == 'E' || *Pstr == 'I' || *Pstr ==
            'O' || *Pstr == 'U')
            count++;
        Pstr++;
    }
    return count;
}
```

Ejercicio N°11

11. Escribe una función llamada "eliminarEspacios" que reciba un puntero a una cadena de caracteres y elimine todos los espacios en blanco que contiene. La función debe modificar la cadena original a través del puntero. En el programa principal, declara una cadena de caracteres, asigna un valor a la cadena, llama a la función "eliminarEspacios" pasando el puntero a la cadena y muestra en pantalla la cadena sin espacios.

```
#include <stdio.h>

void eliminarEspacios(char *cadena);

int main() {
    char cadena[] = "Esta es una cadena con espacios";
    printf("Cadena original: %s\n", cadena);
}
```

```
    eliminarEspacios(cadena);
    printf("Cadena sin espacios: %s\n", cadena);
    return 0;
}

void eliminarEspacios(char *cadena) {
    char *sinEspacios = cadena;
    while (*cadena != '\0') {
        if (*cadena != ' ') {
            *sinEspacios = *cadena;
            sinEspacios++;
        }
        cadena++;
    }
    *sinEspacios = '\0';
}
```

Ejercicio N°12

12. Escribe una función llamada "buscarElemento" que reciba un puntero a un arreglo de enteros, su longitud y un valor entero a buscar. La función debe buscar la primera ocurrencia del valor en el arreglo y retornar la posición (índice) donde se encuentra. Si no se encuentra, la función debe retornar -1. En el programa principal, declara un arreglo de enteros, asigna valores a cada elemento y llama a la función "buscarElemento" pasando el puntero al arreglo y el valor a buscar. Muestra en pantalla la posición encontrada o un mensaje indicando que no se encontró el valor.

```
#include <stdio.h>

int buscarElemento(int *arr, int longitud, int valor);

int main() {
    int array[] = {5, 10, 15, 20, 25};
    int longitud = sizeof(array) / sizeof(array[0]);
    int valorABuscar = 15;
    int posicion = buscarElemento(array, longitud, valorABuscar);
    if (posicion != -1)
        printf("El valor %d se encuentra en la posición %d.\n",
            valorABuscar, posicion);
    else
        printf("El valor %d no se encontró en el arreglo.\n",
            valorABuscar);
    return 0;
}

int buscarElemento(int *arr, int longitud, int valor) {
    for (int i = 0; i < longitud; i++) {
        if (arr[i] == valor)
            return i;
    }
    return -1;
}
```

Ejercicio N°13

13. Escribe una función llamada "encontrarMayor" que reciba un puntero a un número entero y encuentre el valor máximo entre los números almacenados en posiciones de memoria consecutivas. La función debe retornar el valor máximo encontrado. En el programa principal, declara un conjunto de variables enteras y asigna valores a cada una. Llama a la función "encontrarMayor" pasando el puntero a la primera variable y muestra en pantalla el valor máximo.

```
#include <stdio.h>

int encontrarMayor(int *ptr);

int main() {
    int num1[] = {2, 6, 77, 8, 9, 12};
    printf("El valor máximo es: %d\n", encontrarMayor(&num1));
    return 0;
}

int encontrarMayor(int *ptr) {
    int mayor = *ptr;
    for (int i = 0; i < 6; i++){
        if (*ptr > mayor)
            mayor = *ptr;
        ptr++;
    }
    return mayor;
}
```