

# Práctica de punteros

Tecnológico de Costa Rica  
Escuela de Ingeniería en Computadores  
Algoritmos y estructuras de Datos 2  
I Semestre 2025



A continuación se presenta una lista de ejercicios que debe codificar utilizando C/C++. Se recomienda declarar funciones independientes para cada uno de los ejercicios, y llamarlos desde la función *main()*.

1. Escriba un programa que declare e inicialice (a cualquier valor que usted desee) un *double*, un *int* y un *char*. Luego, declare e inicialice un puntero a cada una de las tres variables. Finalmente, su programa debe imprimir la dirección, el valor almacenado y el tamaño de la memoria (en bytes) de cada una de las seis variables.

Utilice el especificador de formato "0x%x" para imprimir direcciones en hexadecimal. Use "%f" para imprimir un valor flotante. Utilice el operador *sizeof()* para determinar el tamaño de memoria (en bytes) asignado para cada variable.

Por ejemplo, la salida del programa debería de verse como:

```
The address of char    ____ is 0x_____  
The address of int     ____ is 0x_____  
The address of double  ____ is 0x_____  
The address of char*   ____ is 0x_____  
The address of int*    ____ is 0x_____  
The address of double* ____ is 0x_____
```

```
The value of char      ____ is _____  
The value of int       ____ is _____  
The value of double    ____ is _____  
The value of char*     ____ is 0x_____  
The value of int*      ____ is 0x_____  
The value of double*   ____ is 0x_____
```

```
The size of char is    _____ bytes  
The size of int is     _____ bytes  
The size of double is  _____ bytes  
The size of char* is   _____ bytes  
The size of int* is    _____ bytes  
The size of double* is _____ bytes
```

2. Cree una función con dos variables enteras, inicializadas a un valor aleatorio, llamadas *int\_1* e *int\_2*, luego cree tres punteros llamados *ptr\_1*, *ptr\_2* y *ptr\_3*. Obtenga la posición de memoria de *int\_1* e *int\_2* y asígnela a *ptr\_1* y *ptr\_2* respectivamente.

Finalmente intercambie el valor de ptr\_1 y ptr\_2 utilizando como pivote a ptr\_3. No obtenga nuevamente la posición de memoria de las variables enteras. Imprima los valores para confirmar el funcionamiento.

3. Declare un puntero sin inicializar e imprima su valor. ¿Qué significa su salida? Ahora intente acceder a esa posición de memoria y explique el resultado.
4. Declare una lista de 25 elementos **enteros** utilizando malloc. Recorra la lista (**utilizando índices**) y asigne sus elementos a un valor aleatorio. Finalmente, recorra la lista otra vez e imprima cada uno de los elementos y confirme que son números aleatorios. Recuerde liberar la memoria.
5. Declare una lista de 25 elementos **enteros** utilizando malloc. Recorra la lista (**utilizando aritmética de punteros**) y asigne sus elementos a un valor aleatorio. Finalmente, recorra la lista otra vez e imprima cada uno de los elementos y confirme que son números aleatorios. Recuerde liberar la memoria.
6. Declare una lista de 25 elementos tipo **double** utilizando malloc. Recorra la lista (**utilizando aritmética de punteros**) y asigne sus elementos a un valor aleatorio. Finalmente, recorra la lista otra vez e imprima cada uno de los elementos y confirme que son números aleatorios. Recuerde liberar la memoria.
7. Los dos ejercicios anteriores utilizaron aritmética de punteros para recorrer dos listas de tipos diferentes. La primera lista contenía números enteros y la segunda contenía números dobles. ¿Por qué si ambos tipos son de tamaños distintos se pueden iterar simplemente haciendo puntero++?
8. Cree una función que reciba un puntero a un número y cámbielo por un número aleatorio. Luego, utilice la función recién creada y confirme que efectivamente el número fue modificado.
9. Cree una estructura que contenga lo siguiente:
  - a. Un número entero
  - b. Un número entero sin signo (unsigned int)
  - c. Un número doble

Luego, cree una función que declare esa estructura usando malloc e inicialice cada uno de los elementos a números aleatorios. Seguido de esto, declare una segunda función que va a recibir un puntero a la estructura. Dentro de esta función imprima los valores originales de la estructura, reemplácelos por otros números aleatorios e imprima la estructura nuevamente.

10. Cree una estructura que contenga lo siguiente:
  - a. Un puntero a una lista de enteros
  - b. Un puntero a una lista de dobles

Luego, reserve memoria para ambas listas suponiendo que cada una es de 25 elementos. Asigne los punteros de las estructuras a los punteros de las dos listas creadas. Finalmente, utilizando los punteros de la estructura, recorra ambas listas y asígneles números aleatorios.