

HI THIS IS BRAIN

Descripción del ejercicio:

Este ejercicio tiene como objetivo trabajar con variables, punteros y referencias en C++ para comprender mejor cómo funcionan. Se tiene que escribir un programa que incluya:

1. Una variable tipo string inicializada con el texto "HI THIS IS BRAIN".
2. Un puntero llamado stringPTR que apunte a esta variable.
3. Una referencia llamada stringREF que haga referencia a la misma variable.

El programa debe imprimir en consola lo siguiente:

1. La dirección de memoria de la variable string.
2. La dirección de memoria almacenada en el puntero stringPTR.
3. La dirección de memoria de la referencia stringREF.

Después, debe mostrar:

1. El valor de la variable string.
2. El valor apuntado por stringPTR.
3. El valor referenciado por stringREF.

No hay funciones prohibidas, y el ejercicio no incluye trucos; simplemente se trata de comprender las referencias como una forma diferente de trabajar con direcciones de memoria.

Propósito del ejercicio:

El objetivo es quitar el misterio a las referencias (&). Aunque tienen pequeñas diferencias con los punteros (*), son una forma más intuitiva de manipular direcciones de memoria y variables.

Cómo afrontar el ejercicio

1. Comprender los conceptos básicos:
 - a. Una variable almacena un valor, como una cadena de texto en este caso.
 - b. Un puntero es una variable que almacena la dirección de memoria de otra variable.
 - c. Una referencia es un alias de una variable existente. Esto significa que no tiene su propia dirección de memoria, pero permite interactuar con la variable original.
2. Estructura el programa:
 - a. Declara una variable de tipo `std::string` y asígnale el texto "HI THIS IS BRAIN".
 - b. Declara un puntero y asígnale la dirección de la variable usando el operador `&`.
 - c. Declara una referencia usando la sintaxis estándar de referencia (`std::string&`).

3. Imprimir la información:

- a. Usando `std::cout` para imprimir las direcciones de memoria. Para esto, recuerda que:
 - i. La dirección de la variable puede obtenerse con `&variable`.
 - ii. Un puntero almacena una dirección de memoria, así que imprímelo directamente.
 - iii. Las referencias no tienen una dirección independiente, pero puedes usar `&` para obtener la dirección de la variable que referencian.
- b. Imprimir los valores:
 - i. El valor de la variable es directamente su contenido.
 - ii. El valor apuntado por el puntero se obtiene con el operador de desreferencia (`*puntero`).
 - iii. El valor referenciado es simplemente el contenido de la referencia.

4. Prueba del código:

- a. Hay que asegurarse que las direcciones impresas para la variable, el puntero y la referencia sean iguales, lo que demuestra que los tres están relacionados con la misma variable.
- b. Se debe verificar que los valores impresos sean idénticos.

Resumen práctico:

1. Declara una variable `std::string`.
2. Usa `&` para inicializar el puntero.
3. Usa `&` para crear una referencia.
4. Imprime las direcciones de memoria de los tres.
5. Imprime los valores usando la variable, el puntero y la referencia.

Este ejercicio ayudará a entender cómo los punteros y referencias permiten interactuar con variables en memoria, haciendo que el manejo de datos sea más flexible y potente.

main.cpp

Explicación clara, sencilla y detallada del código

Este programa está diseñado para demostrar cómo funcionan las variables, punteros y referencias en C++. A continuación, se explica el código paso a paso:

1. Importación de bibliotecas

```
#include <iostream>
#include <string>
```

- `#include <iostream>`: Permite utilizar funciones de entrada/salida como `std::cout` y `std::endl` para imprimir en pantalla.

- `#include <string>`: Habilita el uso de la clase `std::string`, que representa cadenas de texto.

2. Función principal

```
int    main(void) {
```

La función `main` es el punto de inicio del programa. Todo lo que está dentro de esta función se ejecutará.

3. Declaración e inicialización de variables

```
std::string str = "HI THIS IS BRAIN";  
std::string *ptr = &str;  
std::string &ref = str;
```

- `std::string str`: Se declara una variable de tipo `string` y se inicializa con el valor "HI THIS IS BRAIN".
- `std::string *ptr`: Se declara un puntero llamado `ptr`, que almacena la dirección de memoria de la variable `str`. El operador `&` se utiliza para obtener esta dirección.
- `std::string &ref`: Se declara una referencia llamada `ref` que está vinculada directamente a la variable `str`. La referencia actúa como un alias de `str`.

4. Mensajes informativos

```
std::cout << "🧠 Variable, pointer and reference initialized 🧠" << std::endl;
```

Imprime un mensaje para informar que las variables, punteros y referencias se han inicializado.

5. Impresión de direcciones de memoria

```
std::cout << "Address of the string variable: " << &str << std::endl;  
std::cout << "Address held by the ptr pointer: " << ptr << std::endl;  
std::cout << "Address held by the ref reference: " << &ref << std::endl;
```

Aquí se imprimen las direcciones de memoria relacionadas con la variable, el puntero y la referencia:

- `&str`: Obtiene y muestra la dirección de memoria de la variable `str`.
- `ptr`: Muestra directamente la dirección de memoria almacenada en el puntero `ptr`, que debería ser la misma que `&str`.
- `&ref`: Obtiene y muestra la dirección de memoria referenciada por `ref`, que también será igual a `&str`.

6. Impresión de valores

```
std::cout << "Value of the string variable: " << str << std::endl;  
std::cout << "Value pointed to by the ptr pointer: " << *ptr << std::endl;  
std::cout << "Value pointed to by the ref reference: " << ref << std::endl;
```

En esta sección, se imprimen los valores almacenados en las variables:

- `str`: Imprime el valor de la variable, "HI THIS IS BRAIN".
- `*ptr`: Utiliza el operador de desreferencia (*) para acceder al valor almacenado en la dirección de memoria a la que apunta el puntero `ptr`. Este valor también será "HI THIS IS BRAIN".
- `ref`: Imprime directamente el valor referenciado por `ref`, que es "HI THIS IS BRAIN".

7. Finalización del programa

```
return (0);
```

La función `main` retorna 0, indicando que el programa se ejecutó correctamente.

Salida esperada del programa

Al ejecutar el programa, la salida en consola será similar a la siguiente:

🧠 Variable, pointer and reference initialized 🧠

🧠 Printing memory adress 🧠

Address of the string variable: 0x7ffee1bce910
Address held by the ptr pointer: 0x7ffee1bce910
Address held by the ref reference: 0x7ffee1bce910

🧠 Printing values 🧠

Value of the string variable: HI THIS IS BRAIN
Value pointed to by the ptr pointer: HI THIS IS BRAIN
Value pointed to by the ref reference: HI THIS IS BRAIN

En resumen:

1. Direcciones de memoria: El programa muestra que la variable, el puntero y la referencia están asociados a la misma dirección de memoria.
2. Valores: Tanto el puntero como la referencia acceden al mismo valor que la variable `str`.
3. Propósito: Este código demuestra cómo punteros y referencias interactúan con las variables en memoria, facilitando su manipulación desde diferentes enfoques.