Practico - Teoria

En C++, una función es un bloque de código autónomo que realiza una tarea específica. Está diseñado para ser reutilizable y se puede llamar desde cualquier parte del programa. Las funciones pueden tomar parámetros de entrada y devolver valores como salida.
Para definir una función en C++, debe especificar el `tipo de retorno`, `el nombre de la función` y `los parámetros de entrada` (si los hay) en el prototipo de la función. La definición de función contiene el código real que realiza la funcion.
Aquí hay un ejemplo de un prototipo de función y definición que calcula la suma de dos números enteros:

```
// Function prototype
int sum(int a, int b);

int main() {
    return 0;
}

// Function definition
int sum(int a, int b) {
    return a + b;
}
```

En este ejemplo, la función se llama `sum`, toma dos parámetros enteros a y b y devuelve un valor entero. La función simplemente suma los dos enteros de entrada y devuelve el resultado.
Para llamar a la función de suma, simplemente proporcione dos valores enteros como argumentos:

```
int result = sum(5, 3); // result = 8
```

Por tanto, para crear una funcion se debe:

```
return_type function_name(input_type1 arg1, input_type2 arg2, ...) {
    // Function definition
    // Code that performs the task
    // ...
    // Return statement (if return type is not void)
}
```

Aquí hay una guia para crear una función en C++:

```
// Function prototype
int multiply(int a, int b);

int main() {
    // Call the function
    int product = multiply(5, 3); // product = 15
}

// Example function implementation
int multiply(int a, int b) {
    int result = a * b;
    return result;
}
```

En esta guia, debe especificar el prototipo de la función al principio, que incluye el `tipo de retorno`, `el nombre de la función` y `los parámetros de entrada`. En la definición de la función, escribe el código que realiza la tarea y usa la declaración de devolución para devolver el valor de salida (si el tipo de devolución no es `void`).

Las funciones son una parte esencial de `cualquier lenguaje de programación` y le permiten dividir problemas complejos en tareas más pequeñas y manejables. Mediante el uso de funciones, puede escribir un código más `limpio` y `modular`, que es más fácil de leer, comprender y mantener.

Ejercicio

Cree un programa que calcule la suma de los primeros n enteros (es decir, 1 + 2 + ... + n).

- Divida la tarea en funciones más pequeñas: `get_user_input`, `calculate_sum` y `display_output`.
- `get_user_input` debe solicitar al usuario que ingrese un valor para n y lo devuelva como un número entero.
- `calcular_sum` debe tomar un entero n como entrada, calcular la suma de los primeros n enteros y devolverlo como un entero.
- `display_output` debe tomar una suma de enteros como entrada y mostrar el mensaje de: `"La suma de los primeros n enteros es: sum"`.
- Escriba los prototipos de función y las definiciones para cada función:
- Luego, se prueba cada función llamándola con diferentes valores de entrada y verificando si la salida es correcta.
- Finalmente, se documenta el código agregando comentarios que expliquen qué hace el código y cómo funciona.