**PISTAS**

Pista 1: Para trabajar con argumentos variables en C++, necesitarás utilizar la librería **stdarg.h**. Esta librería proporciona una serie de funciones que te permitirán acceder a los argumentos variables de una lista.

Pista 2: Una forma común de trabajar con argumentos variables es utilizar una combinación de funciones como **va\_start**, **va\_arg** y **va\_end**. La función **va\_start** se utiliza para inicializar un objeto de tipo **va\_list**, que te permitirá acceder a los argumentos variables. La función **va\_arg** se utiliza para extraer el siguiente argumento de la lista, mientras que la función **va\_end** se utiliza para finalizar el acceso a la lista.

Aquí te dejo un ejemplo de cómo utilizar estas funciones:

1. #include <stdarg.h>

2.

3. void mi\_funcion(const char\* formato, ...)

4. {

5. va\_list args;

6. va\_start(args, formato);

7.

8. // Aquí puedes utilizar la función va\_arg para extraer los argumentos

9. // de la lista, utilizando el formato para determinar su tipo.

10.

11. va\_end(args);

12. }

13.

Pista 3: Para imprimir texto en la consola, puedes utilizar la función **std::cout** de la librería **iostream**. Esta función es muy similar a la función **printf** en C, pero utiliza una sintaxis ligeramente diferente. Por ejemplo, para imprimir una cadena de texto, puedes utilizar **std::cout << "Hola, mundo!\n";**.

Pista 4: Para recorrer una cadena de texto caracter por caracter en C++, puedes utilizar un bucle **for** con un índice. Por ejemplo:

1. for (int i = 0; formato[i] != '\0'; i++)

2. {

3. // Aquí puedes acceder al carácter en la posición i de la cadena de formato.

4. }

Pista 5: Para detectar si una cadena de formato incluye una determinada secuencia de caracteres, como "%v" en el ejercicio que planteaste, puedes utilizar un condicional dentro del bucle **for** que recorre la cadena de formato. Por ejemplo:

1. for (int i = 0; formato[i] != '\0'; i++)

2. {

3. if (formato[i] == '%' && formato[i+1] == 'v')

4. {

5. // Aquí puedes realizar la acción correspondiente cuando se detecta la secuencia %v.

6. }

7. }

Estas son indicaciones más concretas sobre un posible resultado, aunque no es la única opción:

Creamos la función **mi\_printf\_modificado** que acepta una cadena de formato y un número variable de argumentos, al igual que la función **printf**. Hemos utilizado la librería **stdarg.h** para trabajar con los argumentos variables.

Dentro de la función, hemos recorrido la cadena de formato caracter por caracter, y hemos utilizado un condicional para detectar si el formato incluye la cadena "%v". En caso afirmativo, hemos imprimido el primer argumento de la lista de argumentos variables, y luego hemos recorrido el resto de la lista con un bucle **while**, imprimiendo cada número separado por comas. Hemos utilizado un valor sentinela de -1 para indicar el final de la lista de argumentos.

En caso contrario, si el formato incluye otro tipo de especificador, como "%d" o "%s", hemos utilizado un condicional para detectar el tipo correspondiente, y hemos utilizado la función **va\_arg** para extraer el siguiente argumento de la lista y pasarlo a **std::cout**.

Finalmente, hemos llamado a la función **mi\_printf\_modificado** dos veces, pasándole diferentes argumentos, para demostrar su funcionamiento.

Principio del formulario