Actividad Práctica 2: Expansión de paquetes y sizeof

En esta actividad práctica, exploraremos el uso de la expansión de paquetes y sizeof en plantillas para realizar operaciones simples pero útiles.

Crea una plantilla de función llamada max\_size que acepte un número variable de argumentos de cualquier tipo y devuelva el tamaño en bytes del argumento con mayor tamaño. Si dos argumentos tienen el mismo tamaño, devuelve el primero.

Ejemplo de uso:

1. int main() {

2. std::cout << "El tamaño máximo es: " << max\_size(1, 'a', 3.14, "Hello") << " bytes\n";

3. // Debería imprimir "El tamaño máximo es: 6 bytes"

4. return 0;

5. }

6.

Crea una plantilla de función llamada concat que acepte dos argumentos de cualquier tipo y devuelva un array que contenga los elementos de ambos argumentos. Los argumentos deben tener el mismo tipo. El tamaño del array resultante debe ser la suma de los tamaños de los argumentos.

Ejemplo de uso:

1. int main() {

2. int arr1[] = {1, 2, 3};

3. int arr2[] = {4, 5, 6};

4. auto arr3 = concat(arr1, arr2);

5. for (auto i : arr3) {

6. std::cout << i << ' ';

7. }

8. // Debería imprimir "1 2 3 4 5 6"

9. return 0;

10. }

11.

Crea una plantilla de función llamada sum\_sizeof que acepte un número variable de argumentos de cualquier tipo y devuelva la suma de sus tamaños en bytes.

Ejemplo de uso:

1. int main() {

2. std::cout << "La suma de los tamaños en bytes es: " << sum\_sizeof(1, 'a', 3.14, "Hello") << '\n';

3. // Debería imprimir "La suma de los tamaños en bytes es: 11"

4. return 0;

5. }

6.

Crea una plantilla de función llamada count\_ptrs que acepte un número variable de argumentos de cualquier tipo y devuelva la cantidad de argumentos que son punteros.

Ejemplo de uso:

1. int main() {

2. int x = 5;

3. int\* y = &x;

4. std::cout << "La cantidad de punteros es: " << count\_ptrs(1, 'a', y, "Hello") << '\n';

5. // Debería imprimir "La cantidad de punteros es: 1"

6. return 0;

7. }

8.

Crea una plantilla de función llamada reverse que acepte un número variable de argumentos de cualquier tipo y devuelva los argumentos en orden inverso.

Ejemplo de uso:

1. int main() {

2. auto result = reverse(1, 'a', 3.14, "Hello");

3. std::cout << "En orden inverso: ";

4. (std::cout << ... << result) << '\n';

5. // Debería imprimir "En orden inverso: Hello 3.14 a 1"

6. return 0;

7. }

8.