PISTAS.

1. Pista 1: Asegúrate de que la clase **LinkedList** tenga dos punteros privados a nodos y un contador de elementos.

1. template <typename T>

2. class LinkedList {

3. public:

4. LinkedList() : head(nullptr), tail(nullptr), count(0) {}

5.

6. // ...

7.

8. private:

9. Node<T>\* head;

10. Node<T>\* tail;

11. int count;

12. };

13.

1. Pista 2: Asegúrate de que los métodos de **LinkedList** funcionen correctamente.

1. template <typename T>

2. class LinkedList {

3. public:

4. LinkedList() : head(nullptr), tail(nullptr), count(0) {}

5.

6. void insert(T element) {

7. Node<T>\* newNode = new Node<T>(element);

8. if (head == nullptr) {

9. head = newNode;

10. tail = newNode;

11. } else {

12. tail->next = newNode;

13. tail = newNode;

14. }

15. count++;

16. }

17.

18. void print() {

19. Node<T>\* currentNode = head;

20. while (currentNode != nullptr) {

21. std::cout << currentNode->value << " ";

22. currentNode = currentNode->next;

23. }

24. std::cout << std::endl;

25. }

26.

27. int size() {

28. return count;

29. }

30.

31. private:

32. Node<T>\* head;

33. Node<T>\* tail;

34. int count;

35. };

36.

1. Pista 3: Asegúrate de que la clase **ListPrinter** tenga un constructor que tome un objeto **LinkedList** y un método **print** que imprima los elementos de la lista enlazada.

1. template <typename LinkedListType>

2. class ListPrinter {

3. public:

4. ListPrinter(LinkedListType list) : list(list) {}

5.

6. void print() {

7. list.print();

8. }

9.

10. private:

11. LinkedListType list;

12.

1. };Pista 4: Asegúrate de que los objetos **LinkedList** y **ListPrinter** se creen correctamente en la función **main**.

1. int main() {

2. LinkedList<int> intList;

3. intList.insert(1);

4. intList.insert(2);

5. intList.insert(3);

6.

7. ListPrinter<LinkedList<int>> intPrinter(intList);

8. std::cout << "Int list: ";

9. intPrinter.print();

10. std::cout << "Size: " << intList.size() << std::endl;

11.

12. LinkedList<std::string> stringList;

13. stringList.insert("hello");

14. stringList.insert("world");

15. stringList.insert("!");

16.

17. ListPrinter<LinkedList<std::string>> stringPrinter(stringList);

18. std::cout << "String list: ";

19. stringPrinter.print();

20. std::cout << "Size: " << stringList.size() << std::endl;

21.

22. return 0;

23. }

24.

1. Pista 5: Asegúrate de que los archivos de encabezado necesarios estén incluidos y que no haya errores de sintaxis o de compilación.

1. #include <iostream>

2. #include <string>

3.

4. // ...

5.

6. int main() {

7. // ...

8.

9. return 0;

10. }

11.