6.3c Hint: consider the following idea of a local variable which is treated as global by calls to the internal version of the member function.

```cpp
template <typename Type>
int Binary_search_node::sequence_print( Binary_search_node<Type> *last ) const {
    Binary_search_node<Type> *previous = nullptr;
    sequence_print_internal( previous );
}
```

```cpp
// this member function would be private
template <typename Type>
int Binary_search_node::sequence_print_internal( Binary_search_node<Type> *previous ) const {
    // Your implementation here...
}
```