This code provides a generic linked list implementation. To use it, you first need to define a linked list type using the `___INIT_LIST___` macro:

```
___INIT_LIST___(MyLinkedList, int)
```

This will create a linked list type called `MyLinkedList` that can store integers.

Once you have defined a linked list type, you can create a new linked list using the `create_##LinkedListType()` function:

```
MyLinkedList* list = create_MyLinkedList();
```

You can then add elements to the linked list using the `push_back_##LinkedListType()` or `push_front_##LinkedListType()` functions:

```
push_back_MyLinkedList(list, 10);
push_back_MyLinkedList(list, 20);
push_back_MyLinkedList(list, 30);
```

To search for a node in the linked list based on a lambda function, you can use the `Full_Position_Serch_##LinkedListType()` function. This function takes the following arguments:

- `list`: The linked list to search.
- `Befor`: A pointer to a pointer to the node before the found node.
- `Me`: A pointer to a pointer to the found node.
- `Me_index`: A pointer to an integer that will store the index of the found node.
- `lamda`: A lambda function that takes a pointer to a node as its argument and returns `true` if the node matches the search criteria.
- `...`: Any additional arguments that the lambda function needs.

The following example shows how to use the `Full_Position_Serch_##LinkedListType()` function to search for the first node in the linked list that contains the value `20`:`MyLinkedList* list = create_MyLinkedList()`;

Example usage:

```
#include <stdlib.h>
#include <stdio.h>

#include "linked_list.h"

int main() {
    // Create a new linked list.
    MyLinkedList* list = create_MyLinkedList();

    // Add some elements to the linked list.
    push_back_MyLinkedList(list, 10);
    push_back_MyLinkedList(list, 20);
    push_back_MyLinkedList(list, 30);

    // Search for the first node in the linked list that contains the value `20`.
    MyLinkedListNode* node = NULL;
    int index = Full_Position_Serch_MyLinkedList(list, NULL, &node, NULL,
SearchFor20, 20);

    if (index != -1) {
        // The node containing the value `20` was found at index `index`.
        printf("The node containing the value `20` was found at index %d.\n",
index);
    } else {
        // The node containing the value `20` was not found.
        printf("The node containing the value `20` was not found.\n");
    }

    // Destroy the linked list.
    destroy_MyLinkedList(list);

    return 0;
}
```