Implement methods**add, size**in template class **SLinkedList (which implements List ADT)**representing the singly linked list with type T with the initialized frame. The description of each method is given in the code.

template <class T>  
class SLinkedList {  
public:  
    class Node; // Forward declaration  
protected:  
    Node\* head;  
    Node\* tail;  
    int count;  
public:  
    SLinkedList();  
    ~SLinkedList();  
 void add(T e);  
 void add(int index, T e);  
 int size();  
public:  
    class Node {  
    private:  
        T data;  
        Node\* next;  
        friend class SLinkedList<T>;  
    public:  
        Node() {  
            next = 0;  
        }  
        Node(Node\* next) {  
            this->next = next;  
        }  
        Node(T data, Node\* next) {  
            this->data = data;  
            this->next = next;  
        }  
    };  
};

**For example:**

| **Test** | **Result** |
| --- | --- |
| SLinkedList<int> list;  int size = 10;  for(int index = 0; index < size; index++){  list.add(index);  }  cout << list.toString(); | [0,1,2,3,4,5,6,7,8,9] |
| SLinkedList<int> list;  int size = 10;  for(int index = 0; index < size; index++){  list.add(0, index);  }  cout << list.toString(); | [9,8,7,6,5,4,3,2,1,0] |