## Bài tập CTDLGT – BT Linked List

10 bài đầu trong phần [Linked Lists](https://www.hackerrank.com/domains/data-structures/linked-lists) trên HackerRank.

***Chú ý: Cần thử làm bằng 2 cách lặp và đệ qui.***

### Bài 1 (Solved)

[Print the Elements of a Linked List](https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list/problem)

(In ra các phần từ của một danh sách liên kết)

**Lặp:**

**static** **void** printLinkedList(SinglyLinkedListNode head) {

**while** (head != null)

        {

            System.out.println(head.data);

            head = head.next;

        }

    }

**Đệ quy:**

**static** **void** printLinkedList(SinglyLinkedListNode head) {

**if** (head == null) **return**;

        System.out.println(head.data);

        printLinkedList(head.next);

    }

### Bài 2 (Solved)

[Insert a Node at the Tail of a Linked List](https://www.hackerrank.com/challenges/insert-a-node-at-the-tail-of-a-linked-list/problem)

(Chèn một phần từ vào đuôi danh sách liên kết)

**Lặp:**

**static** SinglyLinkedListNode insertNodeAtTail(SinglyLinkedListNode head, **int** data) {

            SinglyLinkedListNode tmp = **new** SinglyLinkedListNode(data);

            SinglyLinkedListNode itr = head;

**if** (itr == null) **return** tmp;

**while** (itr.next != null) {

                itr = itr.next;

            }

            itr.next = tmp;

**return** head;

    }

**Đệ quy:**

**static** SinglyLinkedListNode insertNodeAtTail(SinglyLinkedListNode head, **int** data) {

**if** (head == null) {

                SinglyLinkedListNode tmp = **new** SinglyLinkedListNode(data);

**return** tmp;

            }

**if** (head.next != null) insertNodeAtTail(head.next, data);

**if** (head.next == null) {

                 SinglyLinkedListNode tmp = **new** SinglyLinkedListNode(data);

                 head.next = tmp;

            }

**return** head;

    }

### Bài 3 (Solved)

### [Insert a node at the head of a linked list](https://www.hackerrank.com/challenges/insert-a-node-at-the-head-of-a-linked-list/problem)

(Chèn một phần từ vào đầu danh sách liên kết)

**static** SinglyLinkedListNode insertNodeAtHead(SinglyLinkedListNode head, **int** data) {

            SinglyLinkedListNode llist = **new** SinglyLinkedListNode(data);

            llist.next = head;

**return** llist;

    }

### Bài 4 (Solved)

### [Insert a node at a specific position in a linked list](https://www.hackerrank.com/challenges/insert-a-node-at-a-specific-position-in-a-linked-list/problem)

(Chèn một phần từ vào vị trí cụ thể trong danh sách liên kết)

**Lặp:**

**public** **static** SinglyLinkedListNode insertNodeAtPosition(SinglyLinkedListNode llist, **int** data, **int** position) {

*// Write your code here*

        SinglyLinkedListNode current = llist;

        SinglyLinkedListNode node = **new** SinglyLinkedListNode(data);

**int** count\_pos = 1;

**while** (count\_pos < position) {

            current = current.next;

            count\_pos++;

        }

        SinglyLinkedListNode tmp = current.next;

        current.next = node;

        node.next = tmp;

**return** llist;

    }

**Đệ quy:**

**public** **static** SinglyLinkedListNode insertNodeAtPosition(SinglyLinkedListNode llist, **int** data, **int** position) {

*// Write your code here*

**if** (position > 1) insertNodeAtPosition(llist.next, data, position-1);

**if** (position == 1) {

            SinglyLinkedListNode node = **new** SinglyLinkedListNode(data);

            SinglyLinkedListNode tmp = llist.next;

            llist.next = node;

            node.next = tmp;

        }

**return** llist;

    }

### Bài 5 (Solved)

[Delete a Node](https://www.hackerrank.com/challenges/delete-a-node-from-a-linked-list/problem)

(Xoá một phần tử)

**Lặp:**

**public** **static** SinglyLinkedListNode deleteNode(SinglyLinkedListNode llist, **int** position) {

*// Write your code here*

*//xoa dau*

**if** (position == 0)

        {

            SinglyLinkedListNode tmphead = llist;

        llist = llist.next;

        tmphead = null;

**return** llist;

        }

*//xoa giua - cuoi*

        SinglyLinkedListNode head = llist;

**int** count = 1;

**while** (count < position) {

            llist = llist.next;

            count++;

        }

        SinglyLinkedListNode tmp = llist.next;

        llist.next = tmp.next;

        tmp = null;

**return** head;

    }

**Đệ quy:**

**public** **static** SinglyLinkedListNode deleteNode(SinglyLinkedListNode llist, **int** position) {

*// Write your code here*

*//xoa dau*

**if** (position == 0)

        {

            SinglyLinkedListNode tmphead = llist;

        llist = llist.next;

        tmphead = null;

**return** llist;

        }

*//xoa giua - cuoi*

**if** (position > 1) deleteNode(llist.next, position-1);

**if** (position == 1) {

            SinglyLinkedListNode tmp = llist.next;

            llist.next = tmp.next;

            tmp = null;

        }

**return** llist;

    }

### Bài 6 (Solved)

[Print in Reverse](https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list-in-reverse/problem)

(In theo thứ tự ngược lại)

**Lặp:**

**public** **static** **void** reversePrint(SinglyLinkedListNode llist) {

*// Write your code here*

        SinglyLinkedListNode before = llist;

        SinglyLinkedListNode center = llist.next;

        SinglyLinkedListNode after = center.next;

**while** (center.next != null) {

            center.next = before;

            before = center;

            center = after;

            after = after.next;

        }

        center.next = before;

        llist.next = null;

**while** (center != null) {

            System.out.println(center.data);

            center = center.next;

        }

    }

**Đệ quy:**

**public** **static** **void** reversePrint(SinglyLinkedListNode llist) {

*// Write your code here*

**if** (llist.next != null) reversePrint(llist.next);

        System.out.println(llist.data);

    }

### Bài 7 (Solved)

[Reverse a linked list](https://www.hackerrank.com/challenges/reverse-a-linked-list)

(Đảo ngược một danh sách liên kết)

**Lặp:**

**public** **static** SinglyLinkedListNode reverse(SinglyLinkedListNode llist) {

        SinglyLinkedListNode before = llist;

        SinglyLinkedListNode center = llist.next;

        SinglyLinkedListNode after = center.next;

**while** (center.next != null) {

            center.next = before;

            before = center;

            center = after;

            after = after.next;

        }

        center.next = before;

        llist.next = null;

**return** center;

    }

**Đệ quy:**

**public** **static** SinglyLinkedListNode reverse(SinglyLinkedListNode llist) {

*//if linked list is null or linked list has only one element*

*//so the reverse of linked list is the original linked list*

**if** (llist == null || llist.next == null) **return** llist;

*//we create a node that hold the head of the linked list*

*//after every recursion*

        SinglyLinkedListNode newHeadNode = reverse(llist.next);

*//in each recursion, we reverse the connection of each elements*

        llist.next.next = llist;

        llist.next = null;

*//return the head (the latter node) of the linked list after*

*//every recursion*

**return** newHeadNode;

    }

### Bài 8 (Solved)

[Compare two linked lists](https://www.hackerrank.com/challenges/compare-two-linked-lists/problem)

(So sánh 2 danh sách liên kết)

**Lặp:**

**static** **boolean** compareLists(SinglyLinkedListNode head1, SinglyLinkedListNode head2) {

**while** (head1 != null && head2 != null) {

**if** (head1.data != head2.data) **return** **false**;

            head1 = head1.next;

            head2 = head2.next;

        }

**if** (head1 != null || head2 != null) **return** **false**;

**return** **true**;

    }

**Đệ quy:**

**static** **boolean** compareLists(SinglyLinkedListNode head1, SinglyLinkedListNode head2) {

**if** (head1 != null && head2 != null){

**if** (head1.data != head2.data) **return** **false**;

        }

**if** ((head1 != null && head2 == null) || (head1 == null && head2 != null))

**return** **false**;

**if** (head1 == null && head2 == null) **return** **true**;

**boolean** result = compareLists(head1.next, head2.next);

**return** result;

    }

### Bài 9 (Solved)

[Merge two sorted linked lists](https://www.hackerrank.com/challenges/merge-two-sorted-linked-lists/problem)

(Hoà trộn hai danh sách liên kết thành một)

**Lặp:**

**static** SinglyLinkedListNode mergeLists(SinglyLinkedListNode head1, SinglyLinkedListNode head2) {

        SinglyLinkedListNode head = null;

        SinglyLinkedListNode itr = null;

**if** (head1.data < head2.data) {

            head = head1;

            itr = head1;

            head1 = head1.next;

        } **else** {

            head = head2;

            itr = head2;

            head2 = head2.next;

        }

**while** (head1 != null && head2 != null) {

**if** (head1.data < head2.data) {

                itr.next = head1;

                itr = itr.next;

                head1 = head1.next;

            } **else** {

                itr.next = head2;

                itr = itr.next;

                head2 = head2.next;

            }

        }

**while** (head1 != null) {

            itr.next = head1;

            itr = itr.next;

            head1 = head1.next;

        }

**while** (head2 != null) {

            itr.next = head2;

            itr = itr.next;

            head2 = head2.next;

        }

**return** head;

    }

**Đệ quy:**

**static** SinglyLinkedListNode mergeLists(SinglyLinkedListNode head1, SinglyLinkedListNode head2) {

        SinglyLinkedListNode head = null;

**if** (head1 == null) head = head2;

**if** (head2 == null) head = head1;

**if** (head1 != null && head2 != null)

        {

**if** (head1.data < head2.data) {

                head = head1;

                head.next = mergeLists(head1.next, head2);

            }

**else** {

                head = head2;

                head.next = mergeLists(head1, head2.next);

            };

        }

**return** head;

    }

### Bài 10

[Get Node Value](https://www.hackerrank.com/challenges/get-the-value-of-the-node-at-a-specific-position-from-the-tail/problem)

(Lấy giá trị)

**Lặp:**

**public** **static** **int** getNode(SinglyLinkedListNode llist, **int** positionFromTail) {

*// Write your code here*

        SinglyLinkedListNode tmp = llist;

**int** length = 0;

**while** (tmp != null) {

            tmp = tmp.next;

            length++;

        }

**int** positionFromHead = length - positionFromTail - 1;

**int** value;

**while** (positionFromHead > 0) {

            llist = llist.next;

            positionFromHead--;

        }

        value = llist.data;

**return** value;

    }

**Đệ quy:**

**public** **static** **int** getNode(SinglyLinkedListNode llist, **int** positionFromTail) {

*// Write your code here*

**int** result = 0;

        SinglyLinkedListNode tmp = llist;

**int** length = 0;

**while** (tmp != null) {

            tmp = tmp.next;

            length++;

        }

**if** (length - positionFromTail - 1 > 0) result = getNode(llist.next, positionFromTail);

**if** (length - positionFromTail - 1 == 0) result = llist.data;

**return** result;

    }