

This challenge is part of a tutorial track by [MyCodeSchool](#) and is accompanied by a video lesson.

You are given the pointer to the head node of a linked list and an integer to add to the list. Create a new node with the given integer. Insert this node at the tail of the linked list and return the head node of the linked list formed after inserting this new node. The given head pointer may be null, meaning that the initial list is empty.

Input Format

You have to complete the `SinglyLinkedListNode insertAtTail(SinglyLinkedListNode head, int data)` method. It takes two arguments: the head of the linked list and the integer to insert at tail. You should **not** read any input from the stdin/console.

The input is handled by code editor and is as follows:

The first line contains an integer n , denoting the elements of the linked list.

The next n lines contain an integer each, denoting the element that needs to be inserted at tail.

Constraints

- $1 \leq n \leq 1000$
- $1 \leq list_i \leq 1000$

Output Format

Insert the new node at the tail and just return the head of the updated linked list. Do **not** print anything to stdout/console.

The output is handled by code in the editor and is as follows:

Print the elements of the linked list from head to tail, each in a new line.

Sample Input

```
5
141
302
164
530
474
```

Sample Output

```
141
302
164
530
474
```

Explanation

First the linked list is NULL. After inserting 141, the list is 141 -> NULL.

After inserting 302, the list is 141 -> 302 -> NULL.

After inserting 164, the list is 141 -> 302 -> 164 -> NULL.

After inserting 530, the list is 141 -> 302 -> 164 -> 530 -> NULL. After inserting 474, the list is 141 -> 302 -> 164 -> 530 -> 474 -> NULL, which is the final list.