

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

You're 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 head of the linked list and return the new head node. The head pointer given may be null meaning that the initial list is empty.

### Input Format

You have to complete the `SinglyLinkedListNode Insert(SinglyLinkedListNode head, int data)` method which takes two arguments - the head of the linked list and the integer to insert. You should NOT read any input from stdin/console.

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

The first line contains an integer  $n$ , denoting the number of elements to be inserted at the head of the list.

The next  $n$  lines contain an integer each, denoting the element to be inserted.

### Constraints

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

### Output Format

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

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

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

### Sample Input

```
5
383
484
392
975
321
```

### Sample Output

```
321
975
392
484
383
```

### Explanation

Initially the list is NULL. After inserting 383, the list is 383 -> NULL.

After inserting 484, the list is 484 -> 383 -> NULL.

After inserting 392, the list is 392 -> 484 -> 383 -> NULL.

After inserting 975, the list is 975 -> 392 -> 484 -> 383 -> NULL.

After inserting 321, the list is 321 -> 975 -> 392 -> 484 -> 383 -> NULL.