

# Insert a node at a specific position in a linked list

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, an integer to add to the list and the position at which the integer must be inserted. Create a new node with the given integer, insert this node at the desired position and return the head node. A position of 0 indicates head, a position of 1 indicates one node away from the head and so on. The head pointer given may be null meaning that the initial list is empty.

## Input Format

You have to complete the `Node* Insert(Node* head, int data, int position)` method which takes three arguments - the head of the linked list, the integer to insert and the position at which the integer must be inserted. You should NOT read any input from stdin/console. `position` will always be between 0 and the number of the elements in the list (inclusive).

## Output Format

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

## Sample Input

```
NULL, data = 3, position = 0
3 --> NULL, data = 4, position = 0
```

## Sample Output

```
3 --> NULL
4 --> 3 --> NULL
```

## Explanation

1. we have an empty list and position 0. 3 becomes head.
2. 4 is added to position 0, hence 4 becomes head.

## Note

For the purpose of evaluation the list has been initialised with a node with data=2. Ignore it, this is done to avoid printing empty lists while comparing output.

## Video lesson