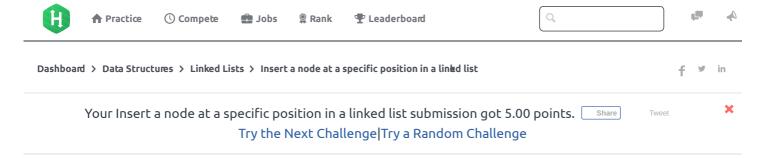
11/06/2017 HackerRank



# 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
```

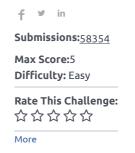
## **Explanation**

- 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



11/06/2017 HackerRank

```
Current Buffer (saved locally, editable) &
                                                                                    C++14
 2
      Insert Node at a given position in a linked list
 3
      head can be NULL
 4
      First element in the linked list is at position \theta
 5
      Node is defined as
 6
      struct Node
 7
 8
          int data;
 9
          struct Node *next;
10
11
12
    Node* InsertNth(Node *head, int data, int position)
13 ▼ {
14
      // Complete this method only
15
      // Do not write main function.
        struct Node *temp,*current;
16
17
         current = head;
18
        int pos = 0;
19
         temp = (struct Node*)malloc(sizeof(struct Node));
20
21
         temp->data = data;
22
         temp->next = NULL;
23
24
         if(position == 0 || head == NULL)
25 ▼
26
                  temp->next = head;
27
                 head = temp;
28
        }
         else
29
30 ▼
31
                  while(pos < position-1)</pre>
32 ▼
                     {
33
                           current = current -> next;
34
                           pos++;
35
36
37
                  temp->next = current->next;
38
                  current ->next = temp;
39
40
         return head;
41
42
   }
43
44
                                                                                                         Line: 31 Col: 35
<u>Upload Code as File</u> Test against custom input
                                                                                                Run Code
                                                                                                            Submit Code
                                     Congrats, you solved this challenge!

✓ Test Case #0

✓ Test Case #1

✓ Test Case #2

✓ Test Case #3

✓ Test Case #4

✓ Test Case #5

✓ Test Case #6

✓ Test Case #7

                                                                                                     Next Challenge
                                                                           You've earned 5.00 points!
```

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature