ESC 101: Fundamentals of Computing		**Minor Quiz 9**		Date: 29– 03 - 2019		
Name						**A**
Roll No.		Dept.		Section		

Instructions:

1. Write final answers neatly with a blue/black pen in the given boxes.
2. **Answers written outside the box will NOT be graded.**

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  struct Node {
4.      int data;
5.      struct Node *next;
6.  };
7.
8.  void print_list(struct Node* head){
9.      while(head){
10.         printf("%d\t",head->data);
11.         head = head->next;
12.     }
13.     printf("\n");
14. }
15.
16. void insert_node(struct Node* node, int data){
17.     struct Node* tmp = node;
18.     node = (struct Node*)malloc(sizeof(struct Node));
19.     node->data = data;
20.     node->next = tmp;
21. }
22.
23. void delete_node(struct Node* node){
24.     struct Node* tmp = node->next->next;
25.     free(node->next);
26.     node->next = tmp;
27. }
28.
29. int main() {
30.     struct Node* head = (struct Node*)malloc(sizeof(struct Node));
31.     head->next = (struct Node*)malloc(sizeof(struct Node));
32.     head->next->next = (struct Node*)malloc(sizeof(struct Node));
33.     head->data = 1;
34.     head->next->data = 2;
35.     head->next->next->data = 3;
36.     print_list(head);
37.     delete_node(head); print_list(head);
38.     insert_node(head, 4); print_list(head);
39.     return 0;
40. }
```

Write the output of the line 36 (2 Marks)

1 2 3

Write the output of the line 37 (2 Marks)

1 3

Does the function insert_node inserts node at the head of the linked list passed in the argument? If yes, write the output of the line 38, if no explain why is it so and correct the code so that insert_node function adds a node to the beginning of the list and write the output of the line 38. (1+5 Marks)

No, because insert_node makes changes to a local pointer to a Node. To implement insert correctly it should take in input a pointer to pointer of the Node and make subsequent changes to value of pointer to pointer variable.

Changes in function definition.

```
void insert_node(struct Node** node, int
data){
    struct Node* tmp = *node;
    *node = (struct
Node*)malloc(sizeof(struct Node));
    (*node)->data = data;
    (*node)->next = tmp;
}
```

Changes in function call.

```
insert_node(&head, 4);
```

Final Output

4 1 3