Major Project: Reverse a LinkedList in O(1) Space Complexity

-by Prakhar Gupta

```
#include <iostream>
using namespace std;
struct node {
   int data; struct node* link;
};
node* rev(node* head) {
   if (head == NULL) cout << "The linklist made is empty\n";
   else if (head->link == NULL) {
      cout << "Deleting the 1 node in linkedlist\n";
      delete head;
      head = NULL;
   }
   else {
      node* previous = NULL;
      node* next = NULL;
      while (head != NULL) {
         next = head->link;
         head->link = previous;
         previous = head;
         head = next;
      }
      head = previous;
   }
   return head;
}
int main() {
   node* head = NULL;
   node* current;
   int data;
   int n;
   cout << "Enter the number of nodes: ";
   cin >> n;
   for (int i = 0; i < n; i++) {
```

```
        current = (struct node*) malloc(sizeof(struct node));
        cout << "Enter the data: ";
        cin >> data;
        current->data = data;
        current->link = head;
        head = current;
    }
    head = rev(head);
    node* p = head;
    while (p != NULL) {
        cout <<" "<< p->data;
        p = p->link;
    }
    return 0;
}
```

Normal linked list:

```
Enter the number of nodes: 5
Enter the data: 5
Enter the data: 4
Enter the data: 3
Enter the data: 2
Enter the data: 1
 1 2 3 4 5
```

Reversed linked list:

```
Enter the number of nodes: 5
Enter the data: 5
Enter the data: 4
Enter the data: 3
Enter the data: 2
Enter the data: 1
 5 4 3 2 1
```