Delete first node of Singly linked list

Given a linked list, the task is to remove the first node of the linked list and update the head pointer of the linked list.

Examples:

```
Input : 1 -> 2 -> 3 -> 4 -> 5 -> NULL
Output : 2 -> 3 -> 4 -> 5 -> NULL

Input : 2 -> 4 -> 6 -> 8 -> 33 -> 67 -> NULL
Output : 4 -> 6 -> 8 -> 33 -> 67 -> NULL
```

To remove first node, we need to make second node as head and delete memory allocated for first node.

```
// CPP program to remove first node of
// linked list.
#include <iostream>
using namespace std;

/* Link list node */
struct Node {
    int data;
    struct Node* next;
};

/* Function to remove the first node
of the linked list */
Node* removeFirstNode(struct Node* head)
{
    if (head == NULL)
        return NULL;

    // Move the head pointer to the next node
    Node* temp = head;
    head = head->next;
```

```
    delete temp;
    return head;
}

// Function to push node at head
void push(struct Node** head_ref, int new_data)
{
    struct Node* new_node = new Node;
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

// Driver codeint main()
{
    /* Start with the empty list */
    Node* head = NULL;

    /* Use push() function to construct
    the below list 8 -> 23 -> 11 -> 29 -> 12 */
    push(&head, 12);
    push(&head, 29);
    push(&head, 11);
    push(&head, 23);
    push(&head, 8);

    head = removeFirstNode(head);
    for (Node* temp = head; temp != NULL; temp = temp->next)
        cout << temp->data << " ";

    return 0;
}
```

Output:

```
23 11 29 12
```

Time complexity : O(1)