Aim

To reverse a singly linked list so that the last node becomes the head and all links are reversed.

Algorithm

1. Initialize three pointers:
    - `prev = NULL`
    - `current = head`
    - `next = NULL`
2. Traverse the list:
    - Store the next node: `next = current->next`
    - Reverse the link: `current->next = prev`
    - Move `prev` forward: `prev = current`
    - Move `current` forward: `current = next`
3. After the loop, set `head = prev`.

C Program

```
#include <stdio.h>

#include <stdlib.h>


// Node structure

struct Node {

    int data;
    struct Node* next;

};


// Function to create new node

struct Node* createNode(int data) {

    struct Node* newNode = (struct Node*)malloc(sizeof(struct
Node));

    newNode->data = data;

    newNode->next = NULL;

    return newNode;

}


// Function to print linked list

void printList(struct Node* head) {

    struct Node* temp = head;

    while (temp != NULL) {

        printf("%d -> ", temp->data);

        temp = temp->next;

    }

    printf("NULL\n");

}


// Function to reverse linked list

struct Node* reverseList(struct Node* head) {

    struct Node* prev = NULL;
```

```c
    struct Node* current = head;
    struct Node* next = NULL;

    while (current != NULL) {
        next = current->next;   // Store next node
        current->next = prev;   // Reverse link
        prev = current;         // Move prev
        current = next;         // Move current
    }
    head = prev;
    return head;
}

int main() {
    // Create linked list: 1 -> 2 -> 3 -> 4 -> NULL
    struct Node* head = createNode(1);
    head->next = createNode(2);
    head->next->next = createNode(3);
    head->next->next->next = createNode(4);

    printf("Original List:\n");
    printList(head);

    head = reverseList(head);
    printf("Reversed List:\n");

    printList(head);


    return 0;

}
```

Input (Hardcoded in Program)

Linked List:

```
1 -> 2 -> 3 -> 4 -> NULL
```

Output

```
Original List:

1 -> 2 -> 3 -> 4 -> NULL

Reversed List:

4 -> 3 -> 2 -> 1 -> NULL
```

Result

The program successfully reverses a singly linked list using iterative pointer manipulation.