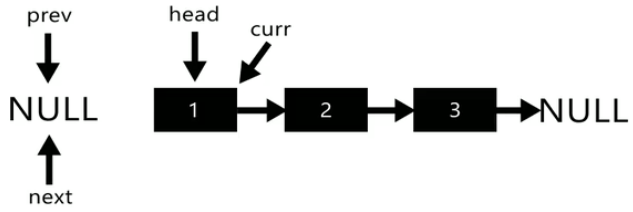
Kodlar

```
#include <stdio.h>
#include <stdlib.h>

/* linked list icin struct olusturulur */
struct Node {
    int data;
    struct Node* next;
};

/* linked listi tersine ceviren fonksiyon */
static void reverse(struct Node** head_ref)
{
    struct Node* prev = NULL;
    struct Node* current = *head_ref;
    struct Node* next = NULL;
    while (current != NULL) {
        // sonraki veri depolanir
        next = current->next;

        // gecerli dugumun isaretcisi onceki degeri gosterir
        current->next = prev;

        // prev isaretcisi gecerli degeri alir, current isaretcisi sonraki degeri alir
        // boylece isaretciler bir adim ilerletilir.
        prev = current;
        current = next;
    }
    *head_ref = prev;
}

/* Yeni dugum ekleme fonksiyonu */
void push(struct Node** head_ref, int new_data)
{
    struct Node* new_node
        = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

/* Linked listi yazdirmak icin olusturdugumuz fonksiyon */
void printList(struct Node* head)
{
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d  ", temp->data);
        temp = temp->next;
```

```
    }
}

int main()
{
    /* Bos bir listeyle basliyoruz */
    struct Node* head = NULL;

    /* Listemizi olusturacak degerleri burda belirliyoruz */
    push(&head, 20);
    push(&head, 4);
    push(&head, 15);
    push(&head, 85);

    printf("Verilen linked list\n");
    printList(head);
    reverse(&head);
    printf("\nTerslenmis Linked list \n");
    printList(head);
    getchar();
}
```

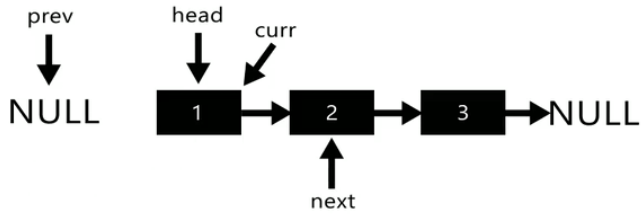
Reverse Fonksiyonu Açıklaması

Başlangıç Durumu



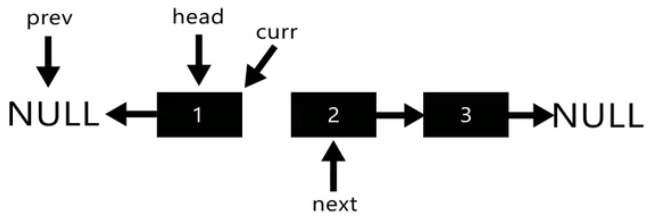

```
while (current != NULL)
    {
        next  = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    *head_ref = prev;
```

1. Durum: next = current->next;



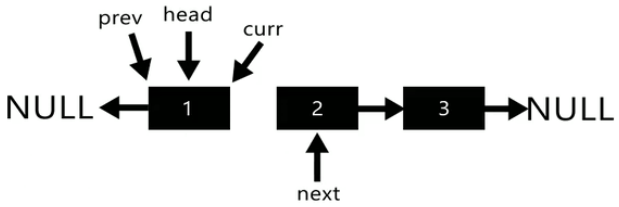

```
while (current != NULL)
     {
          next  = current->next;
          current->next = prev;
          prev = current;
          current = next;
     }
     *head_ref = prev;
```

2. Durum: current->next = prev;



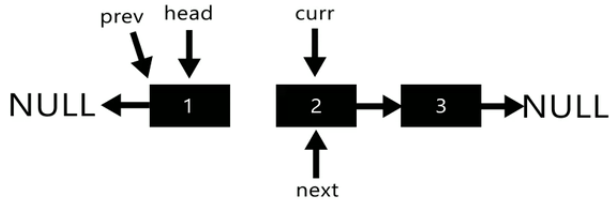

```
while (current != NULL)
     {
          next  = current->next;
          current->next = prev;
          prev = current;
          current = next;
     }
     *head_ref = prev;
```

3. Durum: prev = current;




```
while (current != NULL)
     {
          next  = current->next;
          current->next = prev;
          prev = current;
          current = next;
     }
     *head_ref = prev;
```

4. Durum: current = next;




```
while (current != NULL)
     {
          next  = current->next;
          current->next = prev;
          prev = current;
          current = next;
     }
     *head_ref = prev;
```

Not: Diğer durumlar için döngü bu şekilde devam etmektedir.