Deletion Concept:

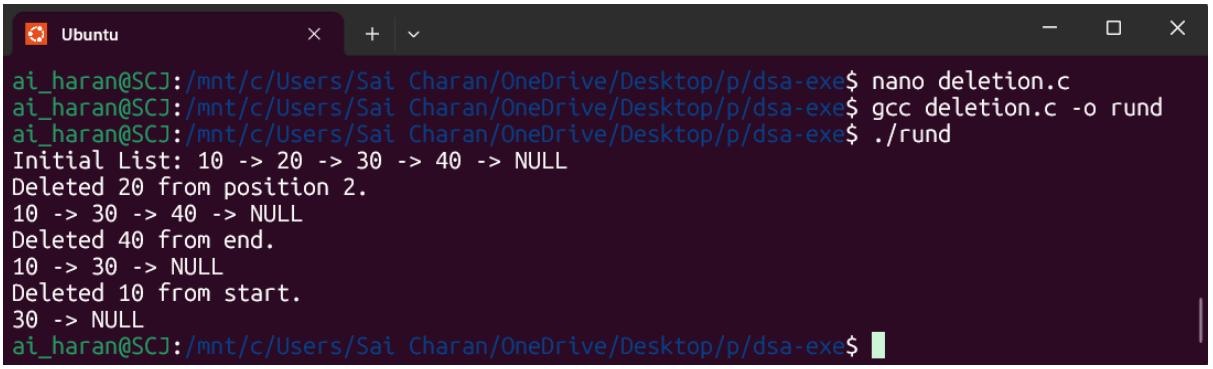
Code:

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

struct Node
{
    int data;
    struct Node *next;
};

struct Node *head = NULL;

// deletion at start
void deleteAtStart()
{
    if (head == NULL)
    {
        printf("List is empty\n");
        return;
    }
    struct Node *temp = head;
    head = head->next;
    printf("Deleted %d from start.\n", temp->data);
    free(temp);
}

// deletion at end
void deleteAtEnd()
{
    if (head == NULL)
    {
        printf("List is empty\n");
        return;
    }
    if (head->next == NULL)
    {
        printf("Deleted %d from end.\n", head->data);
        free(head);
```

```
        head = NULL;
        return;
    }
    struct Node *temp = head;
    while (temp->next->next != NULL)
    {
        temp = temp->next;
    }
    printf("Deleted %d from end.\n", temp->next->data);
    free(temp->next);
    temp->next = NULL;
}

// deletion at specific position
void deleteAtPosition(int position)
{
    if (head == NULL || position < 1)
    {
        printf("Invalid operation\n");
        return;
    }

    if (position == 1)
    {
        deleteAtStart();
        return;
    }

    struct Node *temp = head;

    for (int i = 1; temp != NULL && i < position - 1; i++)
    {
        temp = temp->next;
    }

    if (temp == NULL || temp->next == NULL)
    {
        printf("Position out of range\n");
        return;
    }

    struct Node *delNode = temp->next;
    temp->next = delNode->next;
```

```
    printf("Deleted %d from position %d.\n", delNode->data, position);
    free(delNode);
}

void insert(int val)
{
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->data = val;
    newNode->next = head;
    head = newNode;
}

void display()
{
    struct Node *t = head;
    while (t)
    {
        printf("%d -> ", t->data);
        t = t->next;
    }
    printf("NULL\n");
}

int main()
{
    insert(40);
    insert(30);
    insert(20);
    insert(10);
    printf("Initial List: ");
    display();

    deleteAtPosition(2);
    display();

    deleteAtEnd();
    display();

    deleteAtStart();
    display();

    return 0;
}
```

Output:


Ubuntu
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/dsa-exe$ nano deletion.c
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/dsa-exe$ gcc deletion.c -o rund
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/dsa-exe$ ./rund
Initial List: 10 -> 20 -> 30 -> 40 -> NULL
Deleted 20 from position 2.
10 -> 30 -> 40 -> NULL
Deleted 40 from end.
10 -> 30 -> NULL
Deleted 10 from start.
30 -> NULL
ai_haran@SCJ:/mnt/c/Users/Sai Charan/OneDrive/Desktop/p/dsa-exe$