DSA (CSE102) Tutorial - 6 (Ungraded)

Linked List ,Time :- 45 mins

11-02-25

1 Creating a Node

```c
struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*) _____(sizeof(struct Node)
        );
    newNode->data = _____;
    newNode->next = _____;
    return newNode;
}
```

2 Insertion Operations

2.1 Insert at Beginning

```c
void insertAtBeginning(struct Node** head, int value) {
    struct Node* newNode = _____(value);
    newNode->next = _____;
    *head = _____;
}
```

2.2 Insert at End

```c
void insertAtEnd(struct Node** head, int value) {
    struct Node* newNode = _____(value);
    if (*head == NULL) {
        *head = _____;
        return;
    }
    struct Node* temp = *head;
    while (temp->next != NULL) {
        temp = _____;
    }
    temp->next = _____;
}
```

3 Deletion Operations

3.1 Delete from Beginning

```
void deleteFromBeginning(struct Node** head) {
    if (*head == NULL) return;
    struct Node* temp = _____;
    *head = _____;
    _____(temp);
}
```

3.2 Delete from End

```
void deleteFromEnd(struct Node** head) {
    if (*head == NULL) return;
    if ((*head)->next == NULL) {
        free(*head);
        *head = _____;
        return;
    }
    struct Node* temp = *head;
    while (temp->next->next != NULL) {
        temp = _____;
    }
    _____(temp->next);
    temp->next = _____;
}
```

4 Searching in a Linked List

```
struct Node* search(struct Node* head, int value) {
    struct Node* temp = head;
    while (temp != NULL) {
        if (temp->data == _____) {
            return _____;
        }
        temp = _____;
    }
    return _____;
}
```

5 Traversal in a Linked List

```
void traverse(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", _____);
        temp = _____;
```

```
    }
    printf("NULL\n");
}
```

6 Additional Questions

Write the function which takes head as parameter and performs the following tasks . write individual functions for the following .

6.1 Reversing a Linked List: The TEDx-IIITDelhi Trip to McLeod Ganj

The TEDx-IIITDelhi team embarks on a road trip from IIIT-Delhi to McLeod Ganj, stopping at various locations along the way. However, at the end of the trip, they must retrace their path back to Delhi. Reversing a linked list is similar: we need to change the direction of travel so that the last stop becomes the first.

Example:

```
Before reversal:
IIITD -> Chandigarh -> Dharamshala -> McLeod Ganj -> NULL
After reversal:
McLeod Ganj -> Dharamshala -> Chandigarh -> IIITD -> NULL
```

6.2 Removing Duplicates from a Sorted Linked List: Optimizing the Itinerary

The TEDx-IIITDelhi team planned their trip but accidentally listed some locations multiple times. To optimize their travel, they must remove duplicate stops while keeping the order of travel.

Example:

```
Input: IIITD -> Chandigarh  -> Dharamshala -> McLeod Ganj -> McLeod Ganj -> IIITD -> NULL
Output: IIITD -> Chandigarh -> Dharamshala -> McLeod Ganj -> NULL
```