

WAP to Implement Singly Linked List with following operations a) Create a linked list. b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linked list.

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

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

```
struct Node {
```

```
    int data;
```

```
    struct Node *next;
```

```
};
```

```
struct Node *head = NULL;
```

```
/* Create linked list */
```

```
void createList(int n) {
```

```
    struct Node *newNode, *temp;
```

```
    int data, i;
```

```
    for (i = 0; i < n; i++) {
```

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

```
        printf("Enter data for node %d: ", i + 1);
```

```
        scanf("%d", &data);
```

```
        newNode->data = data;
```

```
        newNode->next = NULL;
```

```
        if (head == NULL) {
```

```
            head = newNode;
```

```
            temp = head;
```

```
        }
```

```
    else {
```

```
        temp->next = newNode;
```

```
        temp = newNode;
```

```
    }
```

```
}
```

```
}
```

```
/* Insert at beginning */
```

```
void insertAtBeginning(int data) {
```

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

```
    newNode->data = data;
```

```

    newNode->next = head;
    head = newNode;
}
/* Insert at end */
void insertAtEnd(int data) {
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    struct Node *temp = head;
    newNode->data = data;
    newNode->next = NULL;
    if (head == NULL) {
        head = newNode;
        return;
    }
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = newNode;
}
/* Insert at any position */
void insertAtPosition(int data, int pos)
{
    struct Node *newNode, *temp = head;
    int i;
    newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->data = data;
    if (pos == 1) {
        newNode->next = head;
        head = newNode;
        return;}
    for (i = 1; i < pos - 1 && temp != NULL; i++) {
        temp = temp->next;
    }
}

```

```

    if (temp == NULL) {
        printf("Invalid position!\n");
        free(newNode);
        return;
    }
    newNode->next = temp->next;
    temp->next = newNode;
}

/* Display linked list */
void displayList()
{
    struct Node *temp = head;
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    int n, choice, data, pos;
    printf("Enter number of nodes to create: ");
    scanf("%d", &n);
    createList(n);
    while (1) {
        printf("\nMenu:\n");
        printf("1. Insert at beginning\n");
        printf("2. Insert at end\n");
    }
}

```

```
printf("3. Insert at any position\n");
printf("4. Display list\n");
printf("5. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
    case 1:
        printf("Enter data: ");
        scanf("%d", &data);
        insertAtBeginning(data);
        break;
    case 2:
        printf("Enter data: ");
        scanf("%d", &data);
        insertAtEnd(data);
        break;
    case 3:
        printf("Enter data: ");
        scanf("%d", &data);
        printf("Enter position: ");
        scanf("%d", &pos);
        insertAtPosition(data, pos);
        break;
    case 4:
        displayList();
        break;
    case 5:
        exit(0);

    default:
        printf("Invalid choice!\n");
}
```

```
}  
  
}
```

```
Enter number of nodes to create: 3  
Enter data for node 1: 10  
Enter data for node 2: 20  
Enter data for node 3: 30  
  
Menu:  
1. Insert at beginning  
2. Insert at end  
3. Insert at any position  
4. Display list  
5. Exit  
Enter your choice: 1  
Enter data: 12  
  
Menu:  
1. Insert at beginning  
2. Insert at end  
3. Insert at any position  
4. Display list  
5. Exit  
Enter your choice: 2  
Enter data: 23  
  
Menu:  
1. Insert at beginning  
2. Insert at end  
3. Insert at any position  
4. Display list  
5. Exit  
Enter your choice: 4  
Linked List: 12 -> 10 -> 20 -> 30 -> 23 -> NULL  
  
Menu:  
1. Insert at beginning  
2. Insert at end  
3. Insert at any position  
4. Display list  
5. Exit  
Enter your choice: 3  
Enter data: 35  
Enter position: 2  
  
Menu:  
1. Insert at beginning  
2. Insert at end  
3. Insert at any position  
4. Display list  
5. Exit  
Enter your choice: 4  
Linked List: 12 -> 35 -> 10 -> 20 -> 30 -> 23 -> NULL  
  
Menu:  
1. Insert at beginning  
2. Insert at end  
3. Insert at any position  
4. Display list  
5. Exit  
Enter your choice: 5  
  
Process returned 0 (0x0)   execution time : 45.552 s  
Press any key to continue.
```

