Singly Linked List Insertion Operations

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

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

struct Node* createNode(int value) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   newNode->next = NULL;
   return newNode;
}

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

void insertAtEnd(struct Node** head, int value) {
   struct Node* newNode = createNode(value);

   if (*head == NULL) {
      *head = newNode;
      return;
   }
```

```
    struct Node* temp = *head;
    while (temp->next != NULL)
        temp = temp->next;

    temp->next = newNode;
}

void insertAtPosition(struct Node** head, int value, int position) {
    if (position == 1) {
        insertAtBeginning(head, value);
        return;
    }

    struct Node* newNode = createNode(value);
    struct Node* temp = *head;

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

    if (temp == NULL) {
        printf("Position out of range!\n");
        return;
    }

    newNode->next = temp->next;
    temp->next = newNode;
}

void display(struct Node* head) {
```

```
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }

    struct Node* temp = head;
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    struct Node* head = NULL;
    int choice, value, position;

    printf("\n--- Singly Linked List Operations ---\n");
    printf("1. Insert at Beginning\n");
    printf("2. Insert at Position\n");
    printf("3. Insert at End\n");
    printf("4. Display List\n");
    printf("5. Exit\n");

    while (1) {
        printf("Enter your choice: ");
        scanf("%d", &choice);
```

```
switch (choice) {
    case 1:
        printf("Enter value to insert: ");
        scanf("%d", &value);
        insertAtBeginning(&head, value);
        break;

    case 2:
        printf("Enter value to insert: ");
        scanf("%d", &value);
        printf("Enter position: ");
        scanf("%d", &position);
        insertAtPosition(&head, value, position);
        break;

    case 3:
        printf("Enter value to insert: ");
        scanf("%d", &value);
        insertAtEnd(&head, value);
        break;

    case 4:
        display(head);
        break;

    case 5:
        exit(0);
```

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

  return 0;
}
```

```
--- Singly Linked List Operations ---
1. Insert at Beginning
2. Insert at Position
3. Insert at End
4. Display List
5. Exit
Enter your choice: 1
Enter value to insert: 10
Enter your choice: 1
Enter value to insert: 9
Enter your choice: 1
Enter value to insert: 8
```

```
Enter value to insert: 8
Enter your choice: 1
Enter value to insert: 6
Enter your choice: 2
Enter value to insert: 7
Enter position: 2
Enter your choice: 3
Enter value to insert: 11
Enter your choice: 4
Linked List: 6 -> 7 -> 8 -> 9 -> 10 -> 11 -> NULL
Enter your choice: 5
 *  Terminal will be reused by tasks, press any key to close it.
```