main.c

Share Run

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

// Define the structure for the linked list node
struct Node {
    int data;
    struct Node* next;
};

// Function to insert a new node at the beginning
void insertAtBeginning(struct Node** head, int newData) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = newData;
    newNode->next = *head;
    *head = newNode;
}

// Function to print the linked list
void printList(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```

Output Clear

```
Linked list after inserting at beginning:
30 -> 20 -> 10 -> NULL


=== Code Execution Successful ===
```

main.c

Share Run Output Clear

```
    *head = newNode;
}

// Function to print the linked list
void printList(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    struct Node* head = NULL; // Start with an empty list

    // Insert nodes at the beginning
    insertAtBeginning(&head, 10);
    insertAtBeginning(&head, 20);
    insertAtBeginning(&head, 30);

    printf("Linked list after inserting at beginning:\n");
    printList(head);

    return 0;
}
```

```
Linked list after inserting at beginning:
30 -> 20 -> 10 -> NULL


=== Code Execution Successful ===
```

main.c

Share | Run | Output | Clear

```c
#include <stdio.h>

int main() {
    int arr[100], n, pos, i;

    // Input size of array
    printf("Enter number of elements: ");
    scanf("%d", &n);

    // Input array elements
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Input position to delete
    printf("Enter the position to delete (1 to %d): ", n);
    scanf("%d", &pos);

    // Check if position is valid
    if(pos < 1 || pos > n) {
        printf("Invalid position!\n");
    } else {
        // Shift elements to left
        for(i = pos - 1; i < n - 1; i++) {
            arr[i] = arr[i + 1];
        }
```

```
Enter number of elements: 3
Enter 3 elements:
3
4
5
Enter the position to delete (1 to 3): 2
Array after deletion:
3 5

=== Code Execution Successful ===
```

main.c

Share Run

```
        scanf("%d", &arr[i]);
    }

    // Input position to delete
    printf("Enter the position to delete (1 to %d): ", n);
    scanf("%d", &pos);

    // Check if position is valid
    if(pos < 1 || pos > n) {
        printf("Invalid position!\n");
    } else {
        // Shift elements to left
        for(i = pos - 1; i < n - 1; i++) {
            arr[i] = arr[i + 1];
        }

        n--;  // Reduce array size

        printf("Array after deletion:\n");
        for(i = 0; i < n; i++) {
            printf("%d ", arr[i]);
        }
    }

    return 0;
}
```

Output Clear

```
Enter number of elements: 3
Enter 3 elements:
3
4
5
Enter the position to delete (1 to 3): 2
Array after deletion:
3 5

=== Code Execution Successful ===
```