```c
#include <stdio.h>
#define MAX 20
struct Node {
    int data;
    struct Node* next;
};
struct Node pool[MAX];
int poolIndex = 0;
struct Node* createNode(int data) {
    if (poolIndex >= MAX) return NULL;
    pool[poolIndex].data = data;
    pool[poolIndex].next = NULL;
    return &pool[poolIndex++];
}
struct Node* createList(int arr[], int size) {
    if (size == 0) return NULL;
    struct Node* head = createNode(arr[0]);
    struct Node* curr = head;
    for (int i = 1; i < size; i++) {
        curr->next = createNode(arr[i]);
        curr = curr->next;
    }
    return head;
}
```

```c
struct Node* insertAtPos(struct Node* head, int p, int n) {
    struct Node* newNode = createNode(p);
    if (newNode == NULL) return head;
    if (n == 1) {
        newNode->next = head;
        return newNode;
    }
    struct Node* temp = head;
    for (int i = 1; i < n - 1 && temp != NULL; i++)
        temp = temp->next;
    newNode->next = temp->next;
    temp->next = newNode;
    return head;
}
void printList(struct Node* head) {
    printf("[");
    while (head) {
        printf("%d", head->data);
        if (head->next) printf(",");
        head = head->next;
    }
    printf("]\n");
}
```

```c
int main() {
    int arr1[] = {1, 3, 2, 3, 4, 5};
    struct Node* head1 = createList(arr1, 6);
    head1 = insertAtPos(head1, 3, 3);
    printf("Output: ");
    printList(head1);
    int arr2[] = {1};
    struct Node* head2 = createList(arr2, 1);
    head2 = insertAtPos(head2, 0, 1);
    printf("Output: ");
    printList(head2);
    int arr3[] = {1, 2};
    struct Node* head3 = createList(arr3, 2);
    head3 = insertAtPos(head3, 3, 3);
    printf("Output: ");
    printList(head3);
    return 0;
}
```

```
Output: [1,3,3,2,3,4,5]
Output: [0,1]
Output: [1,2,3]
```