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

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

void printList(struct Node* head) {
    printf("List: ");
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");
}
int* convertLinkedListToArray(struct Node* head, int* size) {
    int count = 0;
    struct Node* current = head;
    while (current != NULL) {
        count++;
        current = current->next;
    }
    int* array = (int*)malloc(count * sizeof(int));
    if (array == NULL) {
        perror("Memory allocation failed");
        exit(EXIT_FAILURE);
    }
    current = head;
    for (int i = 0; i < count; i++) {
        array[i] = current->data;
        current = current->next;
    }
    *size = count;
    return array;
}
void freeArray(int* array) {
    free(array);
}
void insertAtEnd(struct Node** head, int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    if (newNode == NULL) {
        perror("Memory allocation failed");
        exit(EXIT_FAILURE);
    }
    newNode->data = data;
    newNode->next = NULL;

    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}
int main() {
    struct Node* head = NULL;
    insertAtEnd(&head, 1);
    insertAtEnd(&head, 2);
    insertAtEnd(&head, 3);
    insertAtEnd(&head, 4);
    printList(head);
    int size;
    int* array = convertLinkedListToArray(head, &size);
    printf("Array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");

    freeArray(array);

    return 0;
}
```