

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Node {
5      int data;
6      struct Node* next;
7  };
8
9  void printList(struct Node* head) {
10     printf("List: ");
11     struct Node* current = head;
12     while (current != NULL) {
13         printf("%d ", current->data);
14         current = current->next;
15     }
16     printf("\n");
17 }
18 int* convertLinkedListToArray(struct Node* head, int* size) {
19     int count = 0;
20     struct Node* current = head;
21     while (current != NULL) {
22         count++;
23         current = current->next;
24     }
25     int* array = (int*)malloc(count * sizeof(int));
26     if (array == NULL) {
27         perror("Memory allocation failed");
28         exit(EXIT_FAILURE);
29     }
30     current = head;
31     for (int i = 0; i < count; i++) {
32         array[i] = current->data;
33         current = current->next;
34     }
35     *size = count;
36     return array;
37 }
38 void freeArray(int* array) {
39     free(array);
40 }
41 void insertAtEnd(struct Node** head, int data) {
42     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
43     if (newNode == NULL) {
44         perror("Memory allocation failed");
45         exit(EXIT_FAILURE);
46     }
47     newNode->data = data;
48     newNode->next = NULL;
49
50     if (*head == NULL) {
51         *head = newNode;
52     } else {
53         struct Node* current = *head;
54         while (current->next != NULL) {
55             current = current->next;
56         }
57         current->next = newNode;
58     }
59 }
60 int main() {
61     struct Node* head = NULL;
62     insertAtEnd(&head, 1);
63     insertAtEnd(&head, 2);
64     insertAtEnd(&head, 3);
65     insertAtEnd(&head, 4);
66     printList(head);

```

```
67     int size;
68     int* array = convertLinkedListToArray(head, &size);
69     printf("Array: ");
70     for (int i = 0; i < size; i++) {
71         printf("%d ", array[i]);
72     }
73     printf("\n");
74
75     freeArray(array);
76
77     return 0;
78 }
```