

```

1 //aufgabel.c
2 //Übung 12 Aufgabe 1 GIP
3 //Felix Fleisch Gruppe Die. 14-18 170945
4
5 #include<stdio.h>
6 #include<stdlib.h>
7
8 struct node{
9     int data;
10    struct node* next;
11 };
12
13 void printList(struct node* start){
14     printf("\n");
15     while(start!=NULL){
16         printf("%5d",start->data);
17         start=start->next;
18     }
19     printf("\n");
20 }
21
22 void append(int val,struct node* l){
23     l->next=malloc(sizeof(struct node));
24     l->next->data=val;
25     l->next->next=NULL;
26 }
27
28 void insert(int val,struct node* list){
29     struct node* next=list->next;
30     append(val,list);
31     list->next->next=next;
32 }
33
34 void insertSorted(int val,struct node** start){
35     struct node* p=*start;
36     struct node* zw=*start;
37     if(val>=p->data){
38         while(zw!=NULL){
39             if(zw->data<val){
40                 p=zw;
41             }
42             zw=zw->next;
43         }
44         insert(val,p);
45     }else{
46         struct node* new=malloc(sizeof(struct node));
47         new->data=val;
48         new->next=p;
49         *start=new;
50     }
51 };
52
53 struct node* reverse(struct node* start){
54     //this works
55     int num;
56     struct node* p=start;
57     for(num=1;p->next!=NULL;p=p->next){
58         num++;
59     }
60     void* ptrs[num+1];
61     p=start;
62     int i=0;
63     while(p!=NULL){
64         ptrs[i]=p;
65         i++;
66         p=p->next;
67     }
68     struct node* new=malloc(sizeof(struct node));
69     new->data=((struct node*)ptrs[num-1])->data;
70
71     p=new;
72     printf("test");
73

```

```

74
75     for(int i=num-2;i>=0;i--){
76         append(((struct node*)ptrs[i])->data,p);
77         p=p->next;
78     }
79
80     return(new);
81 };
82
83 int main(){
84     struct node* start=malloc(sizeof(struct node));
85     struct node* list=start;
86     start->data=-10;
87     for(int i=0;i<10;i++){
88         append(i*10,list);
89         list=list->next;
90     }
91     printList(start);
92     int i=0;
93     do{
94         printf("\nBitte Zahl eingeben\n");
95         int n=0;
96         scanf("%d",&n);
97         insertSorted(n,&start);
98         printList(start);
99         printf("\nWeiter? 0/1\n");
100        scanf("%d",&i);
101    }while(i==1);
102    struct node* test=reverse(start);
103    printf("\nUmgekehrt:\n");
104    printList(test);
105    return(0);
106
107 }

```