

Einfache Listen: kopf

```
void durchlaufen(struct knoten* kopf) {
    knoten* laufzeiger = kopf;
    while (laufzeiger != NULL) {
        printf("%d", laufzeiger->wert);
        laufzeiger = laufzeiger->next;
    }
    return 0;
}
```

```
struct knoten* suchen(struct knoten* kopf, int gesuchter_wert) {
    knoten* laufzeiger = kopf;
    if (kopf == NULL) return -1;

    while (laufzeiger != NULL && laufzeiger->wert != gesuchter_wert) {
        laufzeiger = laufzeiger->next;
    }
    return laufzeiger;
}
```

```
/* Liefert -1 im Fehlerfall, sonst 0 */
int einfuegen_kopf(struct knoten* *kopfref, struct knoten* einzufueg) {
    // if (kopfref == NULL) kopfref = einzufueg; //eig unnoetig
    if (einzufueg == NULL) return -1;

    einzufueg->next = *kopfref;
    *kopfref = einzufueg;
    return 0;
}
```

```
int einfuegen_nach(struct knoten* nachdiesem, struct knoten* einzufueg) {
    if(nachdiesem == NULL || einzufueg == NULL) return -1;
    einzufueg->next = nachdiesem->next;
    nachdiesem->next = einzufueg;
    return 0;
}
```

```
int einfuegen_ende(struct knoten* *kopfref, struct knoten* einzufueg) {
    if ( *kopfref == NULL || einzufueg == NULL) return -1;
    if ( *kopfref == NULL) {
        *kopfref = einzufueg;
        einzufueg->next = NULL;
    } else {
        knoten* laufzeiger = *kopfref;
        while(laufzeiger->next != NULL) {
            laufzeiger = laufzeiger->next;
        }
        laufzeiger->next = einzufueg;
        einzufueg->next = NULL;
    }
}
```

kopf,ende

```
void durchlaufen(struct knoten *kopf) {
    struct knoten *laufzeiger;
    laufzeiger = kopf;
    while (laufzeiger != NULL) {
        printf("%d ", laufzeiger->wert);
        laufzeiger = laufzeiger->next;
    }
}
```

```
struct knoten *suchen(struct knoten *kopf, int gesuchter_wert){
    struct knoten *laufzeiger;
    laufzeiger = kopf;
    while ((laufzeiger != NULL) && (laufzeiger->wert != gesuchter_wert)){
        laufzeiger = laufzeiger->next;
    }
    return laufzeiger;
}
```

```
/* Liefert -1 im Fehlerfall, sonst 0 */
int einfuegen_kopf(struct knoten **kopfref, struct knoten **enderef, struct knoten
*einzufueg) {
    if ((einzufueg == NULL) || (kopfref == NULL) || (enderef == NULL))
        return -1;
    einzufueg->next = *kopfref;
    *kopfref = einzufueg;
    return 0;
}
```

```
int einfuegen_nach(struct knoten *nachdiesem, struct knoten *einzufueg, struct
knoten **enderef) {
    if ((nachdiesem == NULL) || (einzufueg == NULL) || (enderef == NULL))
        return -1;
    einzufueg->next = nachdiesem->next;
    nachdiesem->next = einzufueg;
    // ---
    if (einzufueg->next == NULL) {
        *enderef = einzufueg;
    }
    // ---
    return 0;
}
```

```
int einfuegen_ende(struct knoten **kopfref, struct knoten **enderef, struct knoten
*einzufueg) { //O(1)
    if ((einzufueg == NULL) || (kopfref == NULL) || (enderef == NULL)) (*enderef ==
NULL))
        return -1;
    if (*kopfref == NULL) {
        einzufueg->next = NULL;
        *kopfref = einzufueg;
        *enderef = einzufueg;
    } else {
        (*enderef)->next = einzufueg;
        einzufueg->next = NULL;
        *enderef = einzufueg;
    }
    return 0;
}
```

```

/* Liefert den entfernten Knoten oder NULL */
struct knoten* entfernen_kopf(struct knoten* *kopfref) {
    if ((kopfref==NULL)||(*kopfref==NULL)) return NULL;
    knoten* temp = *kopfref;
    temp = temp->next;
    *kopfref = temp;
    return kopfref;
}

```

```

/* Liefert den entfernten Knoten oder NULL */
struct knoten *entfernen_kopf(struct knoten **kopfref, struct knoten **enderef) {
    struct knoten *kopf_alt;
    if ((kopfref == NULL) || (*kopfref == NULL) || (enderef == NULL))
        return NULL;
    kopf_alt = *kopfref;
    *kopfref = (*kopfref)->next;
    return kopf_alt;
}

```

```

struct knoten* entfernen_ende(struct knoten* *kopfref) {

    if ((kopfref==NULL)||(*kopfref==NULL)) return NULL;

    knoten* laufzeiger = *kopfref;
    if(laufzeiger->next == NULL){
        *kopfref = NULL;

        return laufzeiger;
    }

    while ( (laufzeiger->next)->next != NULL){
        laufzeiger = laufzeiger->next;
    }
    laufzeiger->next == NULL;
    return laufzeiger;
}

```

```

struct knoten *entfernen_ende(struct knoten **kopfref, struct knoten **enderef) {
    struct knoten *vor_ende, *ende;
    if ((kopfref == NULL) || (*kopfref == NULL) || (enderef == NULL))
        return NULL;
    if ((*kopfref)->next == NULL) {
        ende = *kopfref;
        *kopfref = NULL;
        return ende;
    }
    vor_ende = *kopfref;
    while (vor_ende->next->next != NULL) {
        vor_ende = vor_ende->next;
    }
    ende = vor_ende->next;
    vor_ende->next = NULL;

    // ---
    *enderef = vor_ende;
    // --

    return ende;
}

```

```

struct knoten* entfernen(struct knoten* *kopfref, struct knoten* auszufueg) {
    knoten* laufzeiger = *kopfref;
    if(*kopfref == NULL || kopfref == NULL || auszufueg == NULL) return -1;

    if (auszufueg == *kopfref) {
        *kopfref = (*kopfref)->next;
        return auszufueg;
    }

    while ((laufzeiger->next) != auszufueg ) {
        if (laufzeiger->next == NULL) return NULL;
        laufzeiger = laufzeiger->next;
    }
    laufzeiger->next = auszufueg->next;
    return auszufueg;
}

```

```

        struct knoten *entfernen(struct knoten **kopfref, struct knoten **enderef,
struct knoten *auszufueg) {
    struct knoten *vor_auszufueg;
    if ((auszufueg == NULL) || (kopfref == NULL) || (*kopfref == NULL) || (enderef ==
NULL))
        return NULL;
    if (auszufueg == *kopfref) {
        *kopfref = (*kopfref)->next;
        return auszufueg;
    }
    vor_auszufueg = *kopfref;
    while (vor_auszufueg->next != auszufueg) {
        if (vor_auszufueg->next == NULL)
            return NULL;
        vor_auszufueg = vor_auszufueg->next;
    }
    vor_auszufueg->next = auszufueg->next;

    // ---
    if (vor_auszufueg->next == NULL) {
        *enderef = vor_auszufueg;
    }
    // ---

    return auszufueg;
}

```