

```

/*
Definizione dei dati
*/
struct Cella {
    int valore;
    struct Cella *next;
};

struct Cella *testa


/*
FUNZIONE ITERATIVA versione 1

esegue l'operazione richiesta sia se la lista è ordinata in modalità crescente
che in modalità decrescente
*/
void eliminaValoriMaggioriK_versione1(struct Cella **t, int k) {
    struct Cella *tmp, *prec, *toDel;

    if (*t == NULL) {
        return;
    }

    prec = NULL;
    tmp = *t;
    while (tmp != NULL) {
        if (tmp->valore > k) {
            toDel = tmp;
            if (prec == NULL) {
                *t = tmp->next;
            }
            else {
                prec->next = tmp->next;
            }
            tmp = tmp->next;

            toDel->next = NULL;
            free(toDel);
        }
        else {
            prec = tmp;
            tmp = tmp->next;
        }
    }
}

```

```

/*
FUNZIONE ITERATIVA versione 2

in questa versione si considera che la lista sia ordinata in modalità crescente
*/
void eliminaValoriMaggioriK_versione2(struct Cella **t, int k) {
    struct Cella *tmp, *prec, *toDel;

    if (*t == NULL) {
        return;
    }

    prec = NULL;
    tmp = *t;
    while (tmp != NULL && tmp->valore <= k) {
        prec = tmp;
        tmp = tmp->next;
    }

    if (tmp == NULL) {
        return;
    }

    if (prec == NULL) {
        *t = NULL;
    }
    else {
        prec->next = NULL;
    }

    while (tmp != NULL) {
        toDel = tmp;
        tmp = tmp->next;

        toDel->next = NULL;
        free(toDel);
    }
}

```

```

/*
FUNZIONE RICORSIVA versione 1

esegue l'operazione richiesta sia se la lista è ordinata in modalità crescente
che in modalità decrescente
*/
struct Cella * funzioneRicorsiva(struct Cella *t, int k) {
    struct Cella *tmp;

    if (t == NULL) {
        return NULL;
    }
    t->next = funzioneRicorsiva(t->next, k);
    if (t->valore > k) {
        tmp = t->next;
        free(t);
        return tmp;
    }
    else {
        return t;
    }
}

```

```
/*
FUNZIONE RICORSIVA versione 2

in questa versione si considera che la lista sia ordinata in modalità crescente
*/
struct Cella * funzioneRicorsiva(struct Cella *t, int k) {
    if (t == NULL) {
        return NULL;
    }
    t->next = funzioneRicorsiva(t->next, k);
    if (t->valore > k) {
        free(t);
        return NULL;
    }
    else {
        return t;
    }
}
```