

```

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

typedef struct listaelem
{
    int adat;
    struct listaelem *kovetkezo;
} LISTAELEM;

LISTAELEM *bovit_elejen(LISTAELEM *f, int ertek)
{
    LISTAELEM *uj = (LISTAELEM *) malloc (sizeof( LISTAELEM ));
    uj->adat = ertek;
    uj->kovetkezo = f;
    return uj;
}

LISTAELEM *bovit_vegen(LISTAELEM *f, int ertek)
{
    LISTAELEM *elozo = NULL, *aktualis = f;
    LISTAELEM *uj = (LISTAELEM *) malloc (sizeof( LISTAELEM ));
    uj->adat = ertek;
    uj->kovetkezo = NULL;
    while(aktualis != NULL)
    {
        elozo = aktualis;
        aktualis = aktualis->kovetkezo;
    }
    if(elozo != NULL)
        elozo->kovetkezo = uj;
    else
        f = uj;
    return f;
}

LISTAELEM *bovit_rendezve(LISTAELEM *f, int ertek)
{
    LISTAELEM *elozo = NULL, *aktualis = f;
    LISTAELEM *uj = (LISTAELEM *) malloc (sizeof( LISTAELEM ));
    while(aktualis != NULL && aktualis->adat < ertek)
    {
        elozo = aktualis;
        aktualis = aktualis->kovetkezo;
    }
    uj->adat = ertek;
    uj->kovetkezo = aktualis;
    if(elozo != NULL)
        elozo->kovetkezo = uj;
    else
        f = uj;
    return f;
}

LISTAELEM *torol_elejrol(LISTAELEM *f)
{
    if(f != NULL)
    {
        LISTAELEM *s = f;
        f = f->kovetkezo;
        free(s);
    }
    return f;
}

LISTAELEM *torol_vegerol(LISTAELEM *f)
{
    LISTAELEM *elozo = NULL, *aktualis = f;
    while(aktualis->kovetkezo != NULL)
    {
        elozo = aktualis;
        aktualis = aktualis->kovetkezo;
    }
    if(elozo != NULL)
        elozo->kovetkezo = NULL;
    else
        f = NULL;
    free(aktualis);
    return f;
}

void bejar(LISTAELEM *f)
{
    LISTAELEM *s = f;
    while(s != NULL)
    {
        printf("%p\tadat: %d\tkov: %p\n",s,s->adat,s->kovetkezo);
        s = s->kovetkezo;
    }
}

int main ()
{
    LISTAELEM *fej = NULL;
    fej=bovit_rendezve(fej,18);
    fej=bovit_rendezve(fej,22);
    fej=torol_vegerol(fej);
    fej=bovit_rendezve(fej,13);
    bejar(fej);
    return 0;
}

```