

Opgave 10 for uge11

I tidligere opgaver med struct person med personer[max] er meget lidt fleksibel hvorimod hvis den bygges op som en hægtet liste så vil listen kunne udvides så den hele tiden passer til det antal records der ligger i filen `const char * filename = "mydatabase.txt";`. Der skal derfor arbejdes med dynamisk allokering af hukommelse som beksrevet i kapitel 12.3 s. 511-512.

Organiseringen af listen skal ske med denne selvreferende node:

```
struct Record {
    char name[50];
    char address[50];
    unsigned int phone;
    struct Record *nextPtr;
};
```

Og det er en fordel at definere synonymer for struct type med: `typedef existing_name alias_name;`

Og synonym for Rekord noden

```
typedef struct Record Record
```

Og synonym for en pointer til Record

```
typedef Record *recordPtr;
```

I mydatabase.txt ligger data komma separeret som

Hans Hansen, Hansvej 45 6700 Esbjerg, 23232323

Jens Jensen, Jensvej 54 5000 Odense, 45454545

Ole Olesen, Grandvej 23 2100 København Ø, 56565656

Denne file er bilag til opgaven

Med udgangspunkt i kapitel 12 s. 513 -521 ønskes der udviklet et program der er i stand til:

1. At indlæse filens data ind i en hægtet liste med overstående node Record i sorteret rækkefølge efter fornavn. Prøv at tegn et flow-chart over hvordan det kan ske inden at du programmerer det.
2. At tilføje en eller flere Records til den sorterede hægtede liste—så det nye element skal ind hvor der hører hjemme i rækkefølgen efter fornavn
3. At den hægtede liste kan skrives ned i en fil i den sorterede orden
4. At man læser filen ind i en sorterede liste ind med Records og så sletter et element og skriver den sorterede liste ned i filen på harddisken igen.