

## Exercise 1 Personregister

Vi ska skapa ett personregister där man kan lägga till personer, ta bort personer, söka efter en person genom namn, skriva ut alla personer och tömma alla personer. Allt ska implementeras med fixed size c-array. Jag implementerade allt med c-array, och 2 pointers. C-array:en kommer att alltid peka på första elementet (startPointer), ena pointern kommer att peka på den sista elementen (endPointer), och andra pointern kommer att peka i början på den första elementen (currentPointer).

När vi lägger ett element så kommer vi sätta datan i nuvarande platsen i currentPointer och därefter kommer currentPointer att inkrementeras så att den pekar alltid på ett steg efter den sista elementet. Med dessa tre pointer så kan vi ta reda på storleken, och kapacitet.

När vi tar bort en element så kommer vi att packa in alla element som kommer efter elementet som ska tas bort ett steg åt vänster. Därefter minskar vi pekaren med minus 1.

Vid sökning itererar vi genom all pointers checkar namnen, ifall namnen matcher så avbryter vi loopen och lämnar tillbaks person pointern, ifall person pointern hittas inte så lämnar vi tillbaka null pointer.

### Print-out

```
-----  
namn: Yousif  
address: Kalle gatan 3a  
namn: Anders  
address: Kalle gatan 1b  
-----
```

## Exercise 2 Fritextsökning

I den här uppgiften ska vi implementera fritextsökning, det vill säga gör det möjligt att söka på en person genom namn eller adress. Man kan också söka efter en viss person genom att mata in en pointer.

Funktionen fungerar genom att vi antingen itererar genom hela arrayen eller så börjar vi iterera igenom startOnNext pointern till slutet av array. Medan vi itererar genom array checkar ifall namnet eller adressen matchar. Ifall det matcher lämnar vi person pointern. Ifall inget hittades så lämnar vi null pointer.

*Print-out*

```
-----  
namn: Yousif  
address: Kalle gatan 3a  
namn: Anders  
address: Kalle gatan 3a  
-----
```

## Exercise 4 Arv

I den här uppgiften skulle vi skapa en subklass av Person klassen som heter PersonMedTel. PersonMedTel ska innehålla ett extra fält telefonnummer, och print metoden overrides så att den loggar också telefonnummer fältet. Därefter ska vi se vad som sker när PersonMedTel används med PersonReg klassen. Ifall PersonRegister klassen innehåller PersonMedTel och vi printar ut PersonReg den kommer att kalla på Person Print metoden och inte PersonMedTel print metoden. Detta sker eftersom PersonMedTel slicas så att den får plats i Person memoryn.

...

*Print-out*

```
-----  
namn: Yousif  
address: Kalle gatan 3a  
namn: Anders  
address: Kalle gatan 1b  
-----
```