Programmering 1

Inlämningsuppgift 10: Filhantering och enumerationer

I denna kurs bestäms 30% av kursvitsordet utgående från resultatet av inlämningsuppgifterna. Denna inlämningsuppgift bidrar med 1/9 av det totala poängantalet. Se kursens hemsidor i Moodle för mera detaljerade bedömningsgrunder.

Ladda upp dina lösningar som en zip-fil till inlämningsmappen i Moodle före genomgångstillfället, dvs. **senast tisdagen den 14.11 kl. 12.30**. Försenade inlämningar ger inga poäng. Efter genomgångstillfället görs modellösningar tillgängliga på Moodle.

1. ***Utökade väderobservationer (100%)***

   Utgå från din egen eller lärarens modellösning för inlämningsuppgift 7.

   Din uppgift är att utöka programmet med ny funktionalitet enligt nedanstående. Implementera den nya funktionaliteten i form av lämpliga funktioner. Sträva som vanligt efter en funktionsindelning som minimerar upprepad kod.

   a) Lägg till en möjlighet att ange vindriktning och vindstyrka för varje observation.

   Vindriktningen skall representeras av en enumeration **`direction_t`** med 8 olika värden för nordlig, nordostlig, ostlig, sydostlig, sydlig, sydvästlig, västlig och nordvästlig vindriktning.

   Vindstyrkan skall representeras som ett heltal.

   Vindriktning och vindstyrka skall tillsammmans representeras som en ny struct **`wind_t`** som i sin tur skall ingå som en ny medlem i den kombinerade structen **`weather_t`**.

   Vid manuell inmatning av väderobservationer kan du låta användaren ange vindriktning genom ett heltal 1..8 (men en mer användarvänlig inmatning är förstås också ok):

   ```
   Wind direction? 2    // 2 motsvarar nordostlig vind
   Wind speed? 7
   ```

   Vid utskrift skall däremot en lämplig strängrepresentation användas (se uppgift b):
   `Wind: NE 2`

   b) Möjliggör *lokalisering* av utskriften av vindriktning genom att läsa in strängrepresentationer för önskat språk från en textfil. Som standard skall engelska användas men andra språk kan väljas genom en *programparameter* när programmet startas, t.ex.

   ```
   ./a.out -l se
   => Wind: NO 2
   ```

(Egentligen borde förstås hela programmet lokaliseras, dvs varenda utskrift borde kunna varieras beroende på språk. Vi väntar dock med detta till kurserna i Objektorienterad programmering och programmeringsspråket Java som har en bra inbyggt API för lokalisering.)

c) Möjliggör inläsning av väderobservationer från en textfil. En textfil skall kunna innehålla ett valfritt antal observationer. Avgör själv hur formatet på datan i textfilen skall se ut.

Ny data skall kunna läsas in genom ett menyval, men användaren skall också kunna ange ett filnamn via en programparameter när programmet startas:

```
./a.out -d oct_2017.data

Read 5 observations.                    // Från oct_2017.data

1. Add observation
2. List all
3. List in range
4. Read from file
5. Exit
4
Enter filename: sept_2017.data

20 observations added.                  // Nu totalt 25 inlästa
```

Situationer där den angivna filen inte kan öppnas eller där innehållet i filen inte motsvarar det förväntade bör hanteras på ett vettigt sätt, dvs utan några krascher eller korrupt data.

d) Inläsning av programparametrar bör implementeras så att parametrarna kan anges i olika ordningsföljd, och så att lämpliga felmeddelanden ges om syntaxen är inkorrekt. Naturligtvis bör också ogiltiga filnamn hanteras korrekt.

```
./a.out -d sept_2017.data -l se       // Två parametrar

./a.out -d
Missing argument to -d

./a.out -x foo
Unknown parameter -x
```