

Enkel in- och utmatning

I denna inledande laboration kommer du att öva på de grundläggande styrstrukturena för repetition och val samt enkel in- och utmatning.

Mål

Du ska efter denna laboration känna till och kunna använda

- hur man skriver ett litet Ada-program
- vad tecknet ”;” betyder och hur det används
- de tre formerna av repetitionssatsen `loop`: `loop`, `for` och `while`
- valsatsen `if`
- läsa in tal med `Get` (dock utan att ta hand om de fel som kan uppstå om användaren skriver in ett heltal då programmet läser ett reellt tal eller tecken)
- `Put` för att skriva ut både heltal, reella tal och strängar både med och utan formatdirektiv.

Uppgift

Skriv och testa momstabellprogrammet. De övriga tre deluppgifterna är frivilliga, men ju fler av dessa du gör desto bättre grund har du att stå på inför resterande laborationskurs.

För samtliga uppgifter gäller att snygga utskrifter ska åstadkommas med hjälp av formatdirektiv till `Put`, samt att du ska kunna motivera valet av iterationssats.

I programmet ska *felhantering* av användarens inmatning göras. Felhanteringen skall vara av typen rimlighetskontroll vilket innebär att användaren alltid kommer att mata in data av rätt datatyp. Detta ska göras så fort som felet *kan* upptäckas. Programmet ska ej avbrytas utan ny fråga ska ställas om användaren matar in ett felaktigt data! Tänk på att användaren kan mata in felaktiga data många gånger ...

Momstabell

Konstruera ett program som skriver ut en momstabell. Programmet ska på terminalen fråga efter och ta som inmatning följande värden (där alla värden ska rimlighetskontrolleras!):

- Nedre samt övre gräns för prisintervallet
- Steglängd i tabellen
- Momsprocenten (uttryckt som decimaltal i intervallet 0 till 100 %)

För att få lite överblick kan det vara vettigt att börja med antagandet att användaren kommer att mata in rimliga värden och skriva ett program utan felkontroller. Programmet ska ge resultat enligt nedan vid körning (användarens indata *kursiverad*):

Exempel 1: Första pris: **10.00**
Sista pris: **15.00**
Steg: **0.5**
Momsprocent: **10.00**

```

=== Momstabell ===
Pris utan moms   Moms      Pris med moms
      10.00      1.00      11.00
      10.50      1.05      11.55
      11.00      1.10      12.10
      .          .          .
      .          .          .
      15.00      1.50      16.50
  
```

Exempel 2: Första pris: **10.00**
Sista pris: **12.00**
Steg: **0.3**
Momsprocent: **20.00**

```

=== Momstabell ===
Pris utan moms   Moms      Pris med moms
      10.00      2.00      12.00
      10.30      2.06      12.36
      10.60      2.12      12.72
      .          .          .
      .          .          .
      11.80      2.36      14.16
  
```

OBS! Sista "Pris utan moms"-värdet i andra exemplet är 11.80 och inte 12.00 (inte heller 12.10)!

Lämpliga testdata kan vara (kombinera lite olika för att se om ditt program verkar fungera):

```

Första pris:    -1   0   10   100
Sista pris:     -1   0   1   11  12  15  101 1000000
Steg:           -1   0   0.1 0.25 0.3 0.5 1 10
Momsprocent:   -1   0   1   20  50  100 101
  
```

Befolkningsproblemet (frivillig)

Befolkningen i två länder antas växa exponentiellt, dvs ökar varje år med givna bråkdelar av befolkningen vid årets början. Skriv ett program som först läser in ett årtal, folkmängden detta år för två länder A och B samt folkökningen i procent för dessa. Programmet skriver därefter ut en tabell med årtal, hur stor folkökningen varit detta år (för varje land), och folkmängd i slutet av året. Detta fortsätter tills det land med från början mindre folkmängd har erhållit en folkmängd som är större än det andra landets. Detta förutsätter att landet med den mindre folkmängden har en större tillväxtprocent än det andra (kontrollera detta när indata matats in).

Programmet ska också besvara frågan "När går det (från början) befolkningsmässigt mindre landet om det större?" och det ska se ut enligt nedan när programmet körs (användarens indata *kursiverad*)

Exempel :

```
Ange startår: 1976
Ange land A:s folkmängd (i milj.): 200
Ange land A:s befolkningsökning (i %): 10.0
Ange land B:s folkmängd (i milj.): 100
Ange land B:s befolkningsökning (i %): 50.0
Årtal  Ökning A  Folkmängd A  Ökning B  Folkmängd B
1976   ---    200.00      ---    100.00
1977  20.00    220.00    50.00    150.00
1978  22.00    242.00    75.00    225.00
1979  24.20    266.20   112.50    337.50
År 1979 har land B gått om land A i befolkning.
```

Födelsedagsproblemet (frivillig)

Beräkna sannolikheten $p(n)$ för att bland n personer åtminstone två har samma födelsedag, samt bestäm det n för vilket $p(n)$ är större än 0.5. Vi bortser från skottår och att nativiteten inte är jämnt fördelad över året.

$$p(n) = 1 - \frac{365}{365} * \frac{364}{365} * \frac{363}{365} \dots \frac{(366-n)}{365}$$

Programmet ska läsa in ett maxvärde på n (exempelvis 100) och sedan skriva ut en tabell över n och $p(n)$ för $n = 1, 2, \dots$ maxvärdet och till sist skriva ut det lägsta n -värde för vilket $p(n)$ är större än 0.5

Kalender (frivillig)

Skriv ett program som skriver ut en kalender för en månad. Utskriften ska se ut så här för en viss månad:

```
Må  Ti  On  To  Fr  Lö  Sö
      1  2  3  4
  5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

Indata till programmet är veckodagsnumret (1 - 7) för första dagen i månaden och antalet dagar som månaden består av. I exemplet ovan är veckodagsnumret 4.

2007-01-29