

## Enkel in- och utmatning

*I denna inledande laboration kommer du att öva på de grundläggande styrstrukturerna 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 %)

Det som är viktigt i denna laboration är att användaren skall känna att den har stöd från programmet om den matar in felaktiga data (felaktiga i detta fall innebär att man matat in ett flyttal, men att det är orimligt givet vad det är man matar in). De körexempel nedan visar hur programmet skall bete sig för just dessa indata. Om man matar in andra data blir givetvis data i tabellen annorlunda, men själva utseendet skall följa den stil som syns i körexemplen.

Det är givet att användaren ALDRIG kommer att mata in något data som är annat än flyttal och dessutom kommer dessa flyttal alltid att ligga i intervallet [-100.00, +100000.00]. Användaren kommer heller aldrig att mata in fler än 2 decimaler. Du skall själv avgöra vad som är rimliga indata för respektive inmatning. Tanken är att användaren skall köpa saker från t.ex. en affär och att affären aldrig kommer att acceptera att den skall betala köparen något.

Ett lämpligt tillvägagångssätt när man programmerar är att man börjar med antagandet att användaren är "snäll" och matar in korrekta och rimliga värden till programmet. Detta gör att man dels slipper hantera felaktiga indata (d.v.s. inga felkontroller behövs i programmet) och dels att man snabbt får fram ett program som "löser" uppgiften.

När man fått sitt program att fungera givet att användaren matar in korrekta data går man vidare till steg två. Detta innebär att man lägger till felhantering av indata. Användaren skall givetvis få sina felmeddelanden så fort det överhuvudtaget är möjligt för att det skall vara bra. Felmeddelanden skall givetvis vara informativa och vänliga så att användaren känner att den är väl omhändertagen.

Följ ovanstående tips även i resterande laborationer så tjänar du en massa tid.

Här följer två programkörningar givet just dessa förutsättningar. Programmet ska ge resultat enligt nedan vid körning (användarens inmatningar markeras nedan genom att texten är *kursiverad* och det som programmet skriver ut är i normalt teckensnitt):

Körexempel 1:

```
INMATNINGSDEL
=====
Mata in första pris: 10.00
Mata in sista pris : 15.00
Mata in steglängd  : 0.5
Mata in momsprocent: 10.00
```

MOMSTABELLEN

```
=====
          Pris                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
```

Körexempel 2:

```
INMATNINGSDEL
=====
Mata in första pris: 10.00
Mata in sista pris : 12.00
Mata in steglängd  : 0.3
Mata in momsprocent: 20.00
```

MOMSTABELLEN

```
=====
          Pris                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)! Detta beror på att det inte går jämnt upp med steglängden givet start- och slutpris. Sista priset skall då inte komma med i tabellen.

Här följer ett körexempel där användaren börjar med att mata in ett felaktigt data. I denna uppgift är det meningen att du skall fundera ut fler fall med orimliga indata(kombinationer) och ge vettiga felmeddelanden för dem. Programmet skall inte avslutas om man matar in felektiga data utan användaren skall få mata in ett korrekt data.

### Körexempel 3:

INMATNINGSDEL

=====

Mata in första pris: **-10.00**

FEL: Första pris måste vara minst 0 (noll) kronor

Mata in första pris: **100.00**

Mata in sista pris : **101.00**

Mata in steglängd : **-10.3**

FEL: Steglängden måste vara minst 0.01

Mata in steglängd : **0.1**

Mata in momsprocent: **12.00**

MOMSTABELLEN

=====

Pris	Moms	Pris med moms
100.00	12.00	112.00
100.10	12.01	112.11
100.20	12.02	112.22
.	.	.
.	.	.
101.00	12.12	113.12

Lämpliga testdata (det finns fler, men detta är en uppsättning att börja med) 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	100000
Steg:	-1	0	0.1	0.25	0.3	0.5	1	10
Momsprocent:	-1	0	1	20	50	100	101	

TIPS: När man testar sitt program är det viktigt att man är noggrann. Man bör skriva upp vilka testfall man vill ha och vad man förväntar sig för resultat. Dessutom bör man testa alla testfall igen om man rättar sitt program när man hittat fel. En bra testare ser "utanför begränsningarna". Det innebär att man ser användaren som en person som inte kanske har förstått hur den skall mata in saker och på det viset kan det bli många roliga testfall.

I denna uppgift har vi dock sagt redan från början att vi håller oss till vissa indata när vi testar era program. Inga värden utanför intervallet [-100.00, 100000.00] kommer att användas och vi anger aldrig fler än 2 decimaler.

## 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
ÅrtalÖkning AFolkmängd AÖkning BFolkmängd 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.

$$p(n)=1 - 365/365 * 364/365 * 363/365 \dots (366-n)/365$$

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.