Tekniska Högskolan i Linköping
Institutionen för Datavetenskap (IDA)
Torbjörn Jonsson 2010-10-22

Snabb introduktion till MatLab

I denna inledande laboration skall du lära känna MatLab som verktyg.

Tidsåtgång

Denna laboration beräknas ta ca 6 timmar i anspråk (4 timmar bokade med assistent).

Mål

Du ska efter denna laboration känna till

- hur man startar och använder MatLab på IDA:s datorer (kanske även varianten där man använder Emacs som editor)
- de grundläggande satserna i MatLab
- hur en funktion ser ut och hur parameteröverföringen går till
- vilka och hur många data som går att returnera från en funktion
- vad det är för skillnad på en funktion och ett script

Uppgift 1 (ca 30 min)

Beräkna summorna/produkten nedan för olika värden på N.

$$\sum_{i=1}^{N} \frac{i^2}{3} \qquad \prod_{i=1}^{N} \frac{1}{i} \qquad \sum_{i=-N}^{N} i\left(\frac{1}{2}\sin(i) + \cos(12i)\right)$$

Du kan lägga alla beräkningarna i samma fil och det skall göras som ett script.

OBS! Det spelar roll med gemener och VERSALER i MatLab.

Uppgift 2 (ca 30 min)

Skriv ett litet program som kontrollerar om ett tal är ett primtal. Tänk på att man vill kunna kontrollera flera olika tal utan att behöva ändra i programmet.

Ett primtal är ett tal som inte går att dela med annat än med 1 och sig själv. Det finns per definition inga primtal som är mindre än talet 2.

Programmet i detta fall skall vara en funktion som tar emot en parameter, det eventuella primtalet, och returnera *true* eller *false*.

Uppgift 3 (ca 45 min)

Skriv en funktion som tar emot ett heltal och returnerar antalet ettor i talet. Exempelvis har talet 11 två ettor, talet 12311 har 3 ettor och talet 234 har 0 ettor.

Tips: Se till att ni inte råkar ut för decimaltal i era beräkningar!

Uppgift 4 (ca 30 min)

Skriv ett progam som låter användaren mata in ett antal sekunder. Låt detta program anropa en funktion som omvandlar dessa till timmar, minuter och sekunder. Spara resultaten i lämpliga variabler. Skriv till sist ut resultatet i programmet.

Gör detta program som ett script!

Uppgift 5 (ca 1 tim (egentligen 00:58:23) :-))

Ett klockslag skrivs som vi alla vet som tre tal separerade med kolon. Varje tal skrivs dessutom med 2 siffror.

Din uppgift är att skriva de instruktioner som krävs för att kontrollera hur lång tid det är mellan två klockslag. Lagra tidpunkt 1 som T1:M1:S1 (d.v.s. T1 motsvarar timmarna etc.) och tidpunkt 2 som T2:M2:S2. Beräkna skillnaden och lagra resultatet i T3:M3:S3 då det är givet att andra tidpunkten är efter den första.

Exempel på klockslag ni kan testa ert program med:

K1	K2	K3
00:00:00	23:59:59	23:59:59
01:02:03	04:05:06	03:03:03
10:23:45	22:22:22	11:58:37
10:10:10	10:10:10	00:00:00
23:59:59	00:00:00	00:00:01

Givetvis skall ert program fråga efter K1 och K2 och användaren skall få ange dessa klockslag. Dessutom skall programmet skriva ut det resulterande tidsintervallet. Inmatningen kan ske så att man matar in timmar, minuter och sekunder separat. Resultatformatet skall vara enligt exemplet nedan.

Tänk till hur du skall lösa problemet innan du skriver programmet. Det krävs endast matematiska operationer. Inga villkor (if) eller upprepningar (for) behövs. Du får inte ens använda dig av upprepningar i just denna uppgift. Du kan tänkas behöva MatLab-funktionerna *fprintf* och/eller *num2str*. Kanske också strängen ’%02d’ (vad kan den betyda?).

Exempel på programkörning:

```
Starttiden:
 Mata in timme : 01
 Mata in minut : 02
 Mata in sekund: 03
Sluttiden:
 Mata in timme : 04
 Mata in minut : 05
 Mata in sekund: 06
Tid mellan start och slut:
 03:03:03
```

Uppgift 6 (ca 45 min)

Skriv ett program som undersöker om ett tal är rikt, fattigt eller perfekt.

Ett tal är rikt om det är positivt och mindre än summan av sina delare.
Ett exempel är 12 < 1 + 2 + 3 + 4 + 6

Ett tal är fattigt om det är positivt och större än sina delare.
Ett exempel är 15 > 1 + 3 + 5

Ett tal är perfekt om det är positivt och lika med summan av sina delare.
Ett exempel är 28 = 1 + 2 + 4 + 7 + 14

Extra frivillig uppgift: Ta reda på vilket som är det 50:e perfekta talet. Denna uppgift ligger (långt) utanför labserien, men kan kanske vara intressant.

Uppgift 7 (ca 2 tim)

Skriv en funktion, no_of_days, som tar emot ett datum och räknar ut vilket dagnummer på året detta datum har. Du skall ta hand om skottår. Första januari har dagnummer 1 och sista december har dagnummer 365 eller 366.

Datumet som kommer till funktionen består av tre heltal motsvarande år, månad och dag.

För att inte ställa till det för er bjuder vi på antalet dagar som finns i respektive månader. Januari, mars, maj, juli, augusti, oktober och december har 31 dagar. Februari har 28 dagar (eller 29 om det är skottår). Övriga har 30 dagar.

Om man använder 2008, 8 och 20 som indata skall man få 233 som resultat (eftersom 2008 är ett skottår).

Ett skottår definieras som ett år som är jämnt delbart med 4. Dock är jämna århundraden inte skottår om de inte är jämnt delbara med 400. Allt detta för att kompensera för att vi inte har exakt 365.25 dagar på ett år. Varför kunde de inte tänkt på det från början? :-)

Krav 1: Du får inte skriva ditt program så att det består av 12 ”if”-grenar för att lösa problemet med månadernas dagantal.

Krav 2: Skottårskontrollen skall göras i en separat funktion som returnerar *true* eller *false*.

Krav 3: Användaren skall få mata in år månad och dag i en funktion som frågar efter dessa data och sen returnerar de tre värdena till det script som agerar huvudprogram.

Tips 1: Användaren matar ALLTID in korrekta data (d.v.s. något som går att tolka som ett datum).

Tips 2: Det är tillåtet att ha fler funktioner.