

## MATLAB-föreläsning 3 (5), 15 september, 2016

## INNEHÅLL

- Upprepning med `for`-slingor och `while`-slingor
- Villkorssatser med `if - then -else - end`
- Logik

## UPPREPNING MED FOR-SLINGA

Ibland vill man utföra samma beräkning eller ett antal satser (kommandoblock) flera gånger. Detta kan göras på olika sätt beroende på vad som styr hur många gånger satserna ska upprepas.

Ett sätt är att använda en så kallad `for`-slinga. Denna använder man om man **på förhand vet hur många gången ett visst kommandoblock ska upprepas**. Generellt så skriver man en `for`-slinga på följande sätt

```
for ii=1:slut
    kommandoblock
end
```

I definitionen av `for`-slingan specificeras att variabeln `ii` ska börja på värdet 1 och anta alla heltal från 1 till `slut`. För varje värde på `ii` utförs kommandona som är specificerade i blocket. En `for`-slinga avslutas alltid med `end`.

Som exempel skriver följande rader ut värdet på en variabel `tal` på skärmen 5 gånger. Värdet på `tal` kommer att ges av variabeln `ii`. Vilka tal kommer vi att se på skärmen?

```
for ii=1:5
    tal=ii
end
```

## UPPREPNING MED WHILE-SLINGA

Om man inte på förhand vet hur många gånger ett kommandoblock måste upprepas utan satserna **upprepas så länge ett villkor är uppfyllt** använder man en `while`-slinga. Generellt skriver man på följande sätt:

```
while avbrottskriterium
    kommandoblock
end
```

I exemplet nedan kommer variabeln `ii` plussas på med ett så länge `ii` är mindre än 3. Vilka tal kommer vi att se på skärmen?

```
ii=0;
while ii<3
    ii=ii+1
end
```

#### NÄSTLADE SLINGOR

Ibland kan man ha användning för en slinga inuti en annan. Det kallas för en **nästlad slinga**. Följande exempel på en nästlad slinga

```
for ii=1:2
    for jj=1:2
        A(ii,jj)=ii*jj
    end
end
```

producerar matrisen  $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ .

**Uppgift 1 - test** Vad skrivs ut på skärmen när följande program, Uppg1.m, körs

```
clear all
a=zeros(1,9); %skapar en radvektor med 9 nollor
for i=1:2:9
    a(i)=i;
end
a
```

**Uppgift 2 - test** Följande rader finns i ett program Uppg2.m. Vilket värde har `x` när det skrivs ut på skärmen?

```
clear all
x=0;
ii=-5;
while ii<0
    x=x+ii;
    ii=ii+1;
end
disp(['x = ' num2str(x)])
```

**Uppgift 3 - test**

Följande rader finns i ett program Uppg3.m. Vad blir utskriften på skärmen när vi kör programmet?

```
clear all
A=[1 2 3;4 5 6; 7 8 9];
for ii=1:3
    for jj=3:-1:1
        disp(['Matrisens element = ' num2str(A(ii,jj))])
    end
end
```

**VILLKORSSATSER**

Om man vill utföra olika beräkningar eller operationer beroende på olika variablers tillstånd kan man använda en if-sats. En if-sats **undersöker om ett villkor är uppfyllt**. Generellt, skriver man en if-sats på följande sätt

```
if (villkor = sant)
    kommandoblock
end
```

Ibland behöver man undersöka flera villkor och utföra kommandon beroende på vad som är uppfyllt och inte. Då använder man en antingen en if/else eller if/elseif/else-sats enligt nedan

```
if villkor 1
    kommandoblock 1 % om villkor 1=sant
else
    kommandoblock 2 % om villkor 1=falskt
end
```

```
if villkor 1
    kommandoblock 1 % om villkor 1=sant
elseif villkor 2
    kommandoblock 2 % om villkor 1=falskt, villkor 2=sant
else
    kommandoblock 3 % om villkor 1=falskt, villkor 2=falskt
end
```

## LOGIK

För att kunna kontrollera om ett villkor är uppfyllt eller inte måste vi testa variabler mot varandra. Detta görs med relationsoperatorerna  $<$  (mindre än),  $>$  (större än),  $<=$  (mindre än eller lika med),  $>=$  (större än eller lika med),  $\sim$  (skiljt från) samt med tre logiska operatorer, NOT ( $\sim$ ), AND ( $\&\&$ ) och OR ( $\|\|$ ). Ett villkor som är uppfyllt (=sant) kommer att evalueras till 1 och ett villkor som inte är uppfyllt (=falskt) evalueras till 0.

**Uppgift 4 - test**

Låt  $a=5$  och  $b=-2$ . Vad blir resultatet av följande kommandorader

- A. 

```
if a>7
    disp(['a = ' num2str(a)])
elseif a<7 && b<0
    disp(['b = ' num2str(b)])
end
```
- B. 

```
if a==5 && b>0
    disp(['a = ' num2str(a)])
elseif a==5 || b>0
    disp(['b = ' num2str(b)])
end
```
- C. 

```
if a~5 && b<0
    disp(['a = ' num2str(a)])
elseif a==5 && b~-2
    disp(['b = ' num2str(b)])
end
```
- D. 

```
if a~5 && b<0
    disp(['a = ' num2str(a)])
elseif a==5 && b~-2
    disp(['b = ' num2str(b)])
else
    disp(['Inget ar ju uppfyllt!!!'])
end
```

**Uppgift 5 - test**

Vad gör följande kod

```
scores=[76, 45, 98, 98];
count=0;
for k=1:length(scores) %length returnerar langden pa vektorn
    if scores(k)>90
        count=count+1;
    end
end
disp(count)
```

**Uppgift 6**

Skriv ett MATLAB-program som omvandlar poängen på en algebratentamen till ett bokstavsbedyg. Betyget ges enligt följande poängfördelning

Betyg	A	B	C	D	E	Fx
Totalt antal poäng	27	24	21	18	16	15
Varav poäng från C	6	3				

Programmet ska ta totala poäng totalt samt poäng på C-delen (där kan man som mest få 12 poäng) som indata och skriva ut betyget på skärmen.