

MATLAB-föreläsning 4+5 (5), 22 och 29 september, 2016

INNEHÅLL

- Villkorssatser med `if - then -else - end` och logik (forts från förra gången)
- Algoritmer och programmeringsstrategi
- Funktioner och anonyma funktioner

Uppgift 1 - test

A. Vilket värde har variabeln `summa` efter följande kommandon?

```
summa=1;
for k=2:4
    summa=summa+k;
end
```

B. Vad skrivs ut på skärmen efter följande kommandon?

```
a=-5;
b=3;
if b<a && a>=0
    c=2*a;
elseif b<a || a<=0
    c=a*b;
else
    c=3*b;
end
c
```

C. Vilket värde har variabeln `antal` när den skrivs ut på skärmen?

```
clear all
nr=[12 11 4 14];
antal=0;
k=0;
while k < length(nr) % funktionen length returnerar
    % längden på vektorn nr
    k=k+1;
```

```
    if nr(k)>10
        antal=antal+1;
    end
end
disp(['Antal = ', num2str(antal)])
```

D. Vad blir utskriften på skärmen?

```
x=[1 2 3 1];
y=[1 2 1 1];
z=[1 1 -1];
if length(x)==length(y) && length(x)==length(z)
    u=x./y+z;
    disp(u)
elseif length(x)==length(y)
    u=x.*y;
    disp(u)
else
    disp('Fel dimension på allt')
end
```

E. Vad blir v efter följande rader av MATLAB-kod?

```
v=[4 3 2 1];

for j=1:length(v)-1
    for i=1:length(v)-1
        if v(i)>v(i+1)
            temp=v(i);
            v(i)=v(i+1);
            v(i+1)=temp;
        end
    end
end
end
v
```

ALGORITMER

En algoritm är en begränsad mängd av väldefinierade instruktioner (kommandon) som givet vissa startdata löser (så effektivt som möjligt) en uppgift med ett ändligt antal steg.

En algoritm består av ett antal rader som utförs sekventiellt (i ordning). I exemplet nedan vill vi hitta det största talet i en vektor.

1. Läs in en vektor med tal.
2. Antag att det första talet i vektorn är det största talet. Beteckna talet med MAX.
3. För resterande tal i vektorn
 - (a) Om talet är större än talet MAX, notera det nya talet som MAX.
 - (b) Om talet är mindre än talet MAX, gå vidare till nästa tal i vektorn.
4. Skriv ut det senast noterade talet MAX (som är det största).

Om vi översätter algoritmen i MATLAB-kod får vi följande

```
clear all
v=input('Ge en vektor med tal');
max=v(1);
for ii=2:length(v)
    if v(ii)>max
        max=v(ii);
    end
end
disp(max)
```

Effektiviteten hos en algoritm bedöms med avseende på

- Hur snabbt den löser ett givet problem
- Hur mycket resurser som krävs (ofta i termer av hur mycket datorminne algoritmen tar i anspråk)
- Hur komplicerad algoritmen är

PROGRAMMERINGSSTRATEGI

- Specificera problemet och vad programmet ska göra.
- Vad är in- och utdata?
- Konstruera en algoritm.
- Omvandla algoritmen till MATLAB-kod.

- Testa programmet.

Uppgift 2

Skriv ett MATLAB-program som omvandlar poängen på en algebratentamen till ett bokstavsbedyg. Maxpoäng på tentamen är 36 poäng. Maxpoäng på C-delen är 12 poäng. Betyget ges enligt följande poängfördelning

Betyg	A	B	C	D	E	F _x
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 som indata och skriva ut betyget på skärmen.

FUNKTIONSFILER

Man kan skriva egendefinierade funktioner i MATLAB. En viktig skillnad mellan en funktion och ett MATLAB-script är att **alla variabler som används i funktionen är avskärmade** från arbetsytan. För att kunna beräkna ett uttryck i en funktion måste man skicka med de värden på variablerna som används i funktionen i ett sk funktionsanrop. Resultatet av beräkningen returneras som utdata från funktionen.

Man skriver en funktion på följande sätt och sparar den i en separat fil med samma namn som funktionsnamnet. I exemplet nedan är funktionen sparad som `minfunktion.m`.

```
function [utdata1, utdata2] = minfunktion(indata1, indata2)
    kommandoblock
end
```

I funktionen ovan har vi specificerat att indata till funktionen är variablerna `indata1` och `indata2`. En funktion kan ha en eller flera indata. På samma sätt har vi specificerat att utdata (det som ska returneras från funktionen när alla kommandon i kommandoblocket är utförda) är variablerna `utdata1` och `utdata2`. En funktion kan ha en eller flera utdata. Både indata och utdata kan vara skalärer, vektorer, matriser eller textsträngar.

Man kan också använda sig av något som kallas för **anonyma funktioner**. Dessa skriver man direkt i sitt MATLAB-program och inte i en separat funktionsfil. En anonym funktion är bra för enklare beräkningar. Om vi vill beräkna funktionen $y(x) = a + bx^2 + cx^3$ för ett antal värden på x kan vi antingen göra det med en anonym funktion i början av det program som ska anropa funktionen

```
y=@(x,a,b,c) a+b*x.^2+c*x.^3;
```

eller skriva en funktionsfil, `yvarde.m`, som innehåller raderna

```
function y = yvarde(x,a,b,c)
    y = a+b*x.^2+c*x.^3;
end
```

Uppgift 3

- A.** Skriv en funktion som beräknar $f(x) = x^2 - 4 \sin 2x - 3$ för x i ett givet intervall. Indata till funktionen ska vara en vektor x och funktionen ska returnera värdena på f för motsvarande värden på x samt min och maxvärdena på f .
- B.** Använd funktionen ovan i ett MATLAB-program som ritar upp funktionen på fyra olika delintervall i x och använder `subplot` för att skapa figuren.
- C.** Antag att vi bara vill veta funktionsvärdena (inte max- och min). Använd en anonym funktion i stället för en funktionsfil.

Uppgift 4

Utöka programmet som beräknar den ortogonala projektionen av en vektor på en linje från föreläsning 2, `Uppg3.m`, så att det även beräknar avståndet från en tredje punkt till linjen. Fundera på om man bör använda en funktion för någon del av programmet.

Uppgift 5 - test

Följande funktion är definierad i filen `avstand.m`.

```
function n = test_fun( x,c,N )
n=1;
while abs(x)<2 && n<N
    x=x^2+c;
    n=n+1;
end
```

Vad blir resultatet av följande anrop

- A.** `>> d=test_fun(1,1,10)`
- B.** `>> d=test_fun(0,1,2)`
- C.** Vad blir resultatet av följande MATLAB-program.

```
clear all
d=test_fun(1,1,10);
disp([' n = ' num2str(n) ])
```