

Skrivning i Matlab-delen av ML1302 Datorbaserade ingenjörswerktyg – Lördag 2018-03-17 LÖSNINGSFÖRSLAG

uppgift 1

a) $[1 \ 3] \cdot [5 \ 6] = [5 \ 18]$

b) Ett enkelt citattecken efter en matris transponerar matrisen. Här kommer radvektorn att omvandlas till en kolumnvektor:

```
>> a'
```

```
ans =
```

```
1
3
```

uppgift 2

$A(:, 3)$ säger att av raderna skall samtliga vara med, och av kolumnerna skall endast kolumn 3 vara med.

$A(3, :)$ säger att av raderna skall endast rad 3 vara med, och av kolumnerna skall alla vara med.

```
>> A(:, 3)
```

```
ans =
```

```
3
6
15
```

```
>> A(3, :)
```

```
ans =
```

```
1    15    15    2
```

uppgift 3

```
>> A = [3 2; -1 4]
```

```
A =
```

```
3    2
-1   4
```

```
>> B = [1 2; 1 3]
```

```
B =
```

```
1    2
1    3
```

2(5)

P är resultatet av en vanlig matrismultiplikation: $A \cdot B = \begin{bmatrix} 3 & 2 \\ -1 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 3 & 10 \end{bmatrix}$

```
>> P=A*B
```

```
P =
```

```
    5    12
    3    10
```

Q är resultatet av att matris A ställs ovanpå matris B .

$$Q = \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

```
>> [A;B]
```

```
ans =
```

```
    3     2
   -1     4
    1     2
    1     3
```

uppgift 4

$t(1)$ blir 1, och sedan ökar $t(n)$ med två för varje varv. Så vi får:

$$t = \begin{bmatrix} 1 & 3 & 5 & 7 \end{bmatrix}$$

$s(n)$ utgörs av summan av alla tidigare t -värden, så vi får:

$$summa = \begin{bmatrix} 1 & 4 & 9 & 16 \end{bmatrix}$$

uppgift 5

```
>> P(:, end)=[]
```

```
P =
```

```
    2    18     6    12
   14    10    13    22
   18     2     3     4
```

Alternativt:

```
>> P(:, 5) = []
```

```
P =
```

```
     2     18     6     12
    14     10    13     22
    18      2      3      4
```

Ytterligare alternativ:

```
>> P = P(:, 1:4)
```

```
P =
```

```
     2     18     6     12
    14     10    13     22
    18      2      3      4
```

uppgift 6

Punkterna är representerade av vektorer. Punkt 1 representeras av $r_1 = (x_1, y_1)$ och punkt 2 av $r_2 = (x_2, y_2)$. Funktionen blir:

```
function d=avstand(r1, r2)
x1=r1(1);
x2=r2(1);
y1=r1(2);
y2=r2(2);

d=sqrt( (x1-x2)^2 + (y1-y2)^2 );
```

Alternativ lösning:

```
function d=avstand(r1, r2)
d = sqrt(sum((r1-r2).^2));
```

uppgift 7

mat-filer är ett slags binärfiler där man kan spara data (matriser och andra variabler). För att, exempelvis, spara variabeln **x** i filen **SparatX.mat** kan man skriva:

```
save SparatX.mat x
```

mat-filen skapas när man använder kommandot **save** på detta sätt. För att spara matrisen **M** och textsträngen **s** i filen **MatStrng.mat** kan man skriva:

```
save MatStrng.mat M s
```

För att spara alla definierade variabler i filen **AllaVariabler.mat** kan man skriva:

```
save AllaVariabler.mat
```

För att åter få tillgång till variablerna i MATLAB-miljön används kommandot **load**:

```
load AllaVariabler.mat
```

uppgift 8

\mathbf{x} är en radvektor med 1200 element. Därför blir också $\exp(-\mathbf{x}/3)$ en radvektor. Både \mathbf{x} och $\exp(-\mathbf{x}/3)$ kan också betraktas som en matris med en rad och 1200 kolumner. Detta gäller också $\sin(\mathbf{x})$. Så uttrycket $\exp(-\mathbf{x}/3) * \sin(\mathbf{x})$ tolkas av MATLAB som en matrismultiplikation, som emellertid inte är tillåten. En matrismultiplikation kräver ju att antalet kolumner i den vänstra matrisen är lika med antalet rader i den högra matrisen. Detta är inte uppfyllt här.

För att åstadkomma det vi vill skall vi använda oss av elementvis multiplikation (operatör `'.*'`):

```
y=exp(-x/3).*sin(x);
```

uppgift 9

En vektor med tärningskast kan skapas med:

```
v=floor(rand(1, 20)*6)+1
```

Den efterfrågade funktionen kan vara:

```
function t=TarningsProtokoll(v)

N=length(v);
t=zeros(1, 6);
for n=1:N
    t(v(n)) = t(v(n)) + 1;
end
```

Ett skript som anropar funktionen skulle kunna se ut så här:

```
v=floor(rand(1, 20)*6)+1

P=TarningsProtokoll(v);

fprintf('\n');
fprintf('Antal ettor: %d\n', P(1));
fprintf('Antal tvåor: %d\n', P(2));
fprintf('Antal treor: %d\n', P(3));
fprintf('Antal fyror: %d\n', P(4));
fprintf('Antal femmor: %d\n', P(5));
fprintf('Antal sexor: %d\n', P(6));
```

uppgift 10

Här är en lösning som klarar bokstäverna **a-z**:

```
s = input('Mata in textsträng: ', 's');

L=length(s);
small = s;
CaseDiff = 'a' - 'A';

for k=1:L
    if s(k)>='A' & s(k)<='Z'
        small(k) = s(k) + CaseDiff;
    end
end
fprintf('%s\n', small);
```

Vill man klara av övriga bokstäver i alfabetet kan man komplettera koden. Här är det gjort endast för bokstaven **å**, men samma princip gäller för de andra bokstäverna:

```
s = input('Mata in textsträng: ', 's');

L=length(s);
small = s;
CaseDiff = 'a' - 'A';

for k=1:L
    if s(k)>='A' & s(k)<='Z'
        small(k) = s(k) + CaseDiff;
    end
    if s(k) == 'Å'
        small(k) = 'å';
    end
end
fprintf('%s\n', small);
```

Den enklaste lösningen är att använda MATLABs funktion **lower**:

```
s = input('Mata in textsträng: ', 's');

small=lower(s);
fprintf('%s\n', small);
```

Men då är det nästan så att man slipper programmera.