

Datorbaserade ingenjörswerktyg, ML1302

KON1: Kontrollskrivning i programmering med MATLAB

Torsdag 2018-06-07

LÖSNINGSFÖRSLAG

uppgift 1

a)

x blir 182.99, dvs. talet avrundas till två decimaler.

b)

y blir **ABCD**

Med kolon-kommando skapar man en talföljd eller en teckenföljd (dvs. en teckensträng). I detta fall blir det en teckensträng som börjar med **A**, slutar med **D** och som innehåller de mellanliggande bokstäverna.

Användning av funktionen **char** garanterar att man får **ASCII**-tecken.

uppgift 2

varv 1: k är 1 och r blir: $r + k = 0 + 1 = 1$

varv 2: k är 3 och r blir: $r + k = 1 + 3 = 4$

varv 3: k är 5 och r blir: $r + k = 4 + 5 = 9$

varv 4: k är 7 och r blir: $r + k = 9 + 7 = 16$

Svar:

Efter **for**-slingan gäller:

k = 7

r = 16

uppgift 3

```
x=linspace(-1, 1, 2000);  
y=sqrt(1-x.*x).*sin(40*x);  
plot(x, y);  
grid on;
```

Det är mycket viktigt med punkterna på den andra raden.

uppgift 4

```
B=A(1:2, 1:3)
```

uppgift 5

```
function ut=heltaltest(x)
if fix(x)~=x
    ut = 0;
else
    ut = 1;
end
```

Alternativ lösning:

```
function ut=heltaltest2(x)
ut=1;
if fix(x)~=x
    ut = 0;
end
```

uppgift 6

a) **??? Index exceeds matrix dimensions.**

Man har försökt referera till ett element i en vektor eller matris på ett felaktigt sätt. Antingen har man använt ett för högt index, och försökt referera till ett element som inte existerar, eller refererat med ett icke-positivt index. Om en vektor \mathbf{v} har tio element blir det fel om man gör referensen $\mathbf{v}(11)$ och det blir självklart fel om man skriver $\mathbf{v}(0)$ eller $\mathbf{v}(-4)$.

b) **??? Error using ==> plus
Matrix dimensions must agree.**

Vid addition mellan matriser måste antalet rader och kolumner i den vänstra matrisen stämma överens med antalet rader och kolumner i den högra matrisen. Exempel: om en 5×4 -matris skall adderas till en annan matris måste den andra matrisen också vara en 5×4 -matris. Annars erhålls detta felmeddelande.

uppgift 7

```
function ovr10=noverten(A)

[nrad, nkol]=size(A);
ovr10=0;

for r=1:nrad
    for k=1:ncol
        if A(r, k)>=10
            ovr10=ovr10+1;
        end
    end
end
```

Alternativ lösning:

```
function ovr10=noverten2(A)

ovr10=sum(sum(A>=10));
```

uppgift 8

```
>> global g=9.81
>> [t, y]=FallingMass(80);
>> plot(t, y)
```

uppgift 9

```
function trimmed=NumberTrimmer(s)
N=length(s);
trimmed=[];
m=0;
for n=1:N
    if (s(n)>='0') && (s(n)<='9')
        m=m+1;
        trimmed(m)=s(n);
    end
end
trimmed=char(trimmed);
```

Med ovanstående lösning utökas antalet element i `trimmed` allteftersom nya siffror läggs till strängen. Detta anses inte riktigt rumsrent av MATLAB som ger varningsmeddelande för detta. Vi vet att `trimmed` högst kan ha samma antal element som `s`. Så vi kan i början låta `trimmed` ha samma antal element som `s` och sedan skära bort de sista elementen när sifferföljden är klar. Då blir koden så här:

```
function trimmed=NumberTrimmer2(s)
N=length(s);
trimmed=s;
m=0;
for n=1:N
    if (s(n)>='0') && (s(n)<='9')
        m=m+1;
        trimmed(m)=s(n)
    end
end
trimmed=trimmed(1:m);
trimmed=char(trimmed);
```

uppgift 10

```
function starring(N)

for n=1:N
    fprintf('* ');
end
fprintf('\n');
```

Alternativ:

```

function starring2(N)

s=ones(1,N);
for n=1:N
    s(n)='*';
end
fprintf('%s\n', s);

```

Ytterligare alternativ:

```

function starring3(N)

disp(char('*'*ones(1, N)) );

```