

# Klausur Grundlagen Informatik – Musterklausur & Lösungen

(Die Lösungen sind der Klausur angefügt.)

Name:

Matrikelnummer:

A1	A2	A3	A4	Σ

## Aufgabe 1 (10 Punkte, 2 pro Teilaufgabe)

a) Worin unterscheiden sich Datentypen?

**Lösung:** \_\_\_\_\_

b) Übersetzen Sie folgende Formel in einen Matlab-Ausdruck.

$$y = \begin{cases} \pi \frac{x}{2} & \text{falls } x < 0 \\ 0 & \text{falls } x = 0 \\ \frac{\pi}{3} x & \text{falls } x > 0 \end{cases}$$

**Lösung:** \_\_\_\_\_

c) Geben Sie die einzeilige MATLAB-Notation für folgenden mathem. Ausdruck an

$$z + \left( \sin(x) + \sqrt[3]{\frac{\log^2(y)}{(2 + \sqrt{z})^3}} \right)$$

**Lösung:** \_\_\_\_\_

d) Welches Ergebnis liefert der MATLAB Ausdruck  $\mathbf{x} \geq \mathbf{y} > \mathbf{z}$

a) für  $x=1, y=1, z=1$

b) für  $x=1, y=1, z=0$

**Lösung:** \_\_\_\_\_

e) Geben Sie die Ausgaben der folgenden Codestücke an.

<p><b>(e1)</b></p> <pre>for i =1:5     disp(i) end</pre>	<p><b>(e2)</b></p> <pre>i=0; while i&lt;5     i=i+1;     disp(i) end</pre>	<p><b>(e3)</b></p> <pre>i=0; while i&lt;=5     disp(i)     i=i+1; end</pre>
--	--	---

(e1) **Lösung:** \_\_\_\_\_

(e2) **Lösung:** \_\_\_\_\_

(e3) **Lösung:** \_\_\_\_\_

## Aufgabe 2 (10 Punkte, 2 pro Teilaufgabe)

a) Welches Ergebnis liefern folgende Ausdrücke für  $v=[1.1, 2.2, 3.3]$  und  $w=[3, 2, 1]$ . Falls ein Ausdruck einen Fehler erzeugt, geben Sie den Fehler an.

>>  $w(v(3))$                       -> **Lösung:** \_\_\_\_\_  
 >>  $v([w(1), w(3)])$             -> **Lösung:** \_\_\_\_\_

b) Geben Sie den einzeiligen MATLAB Ausdruck zur Berechnung folgender Formel an, wenn  $x=[1.1, 2.2, 3.3; 10.1, 20.2, 30.3]$  ist. Welche Dimension hat  $y$ ?

$$y = \sin(x \cdot (x^2 + 3x + 1))$$

**Lösung:** \_\_\_\_\_  
 \_\_\_\_\_

Suchen Sie die Stellen in folgenden Funktionen, die eine Fehler- oder Warnmeldung verursachen. **Markieren** (hervorheben oder einkreisen) Sie die Zeilen mit einem Fehler und **begründen** Sie Ihre Entscheidung (z.B. Übergabeparameter sind skalar).

c) `function x = f2(b)`

```
if (cos(b)- sin(b+1))*b > 0
    x = 1
else
    x = 0
end
```

**Lösung (Begründung):** \_\_\_\_\_

d) `function y = f4(a)`

```
v = [a; pi];
if v(2) > round(v(1))
    y = v(1);
else
    y = pi + v(3);
end
```

**Lösung (Begründung):** \_\_\_\_\_

e) Programmanalyse: Welches Ergebnis liefert die folgende Funktion bei den angegebenen Aufrufen? Bei Fehlern beschreiben Sie kurz das Problem.

```
function v = f2(x,n)
v = zeros(n,1);
for i = 1 : round(n/2)+x
    v(i) = i * x;
end
```

(e1) >>f2(2,-1) ->**Lösung:** \_\_\_\_\_  
 (e2) >>f2(1,4) ->**Lösung:** \_\_\_\_\_

**Lösung (Probleme):** \_\_\_\_\_  
 \_\_\_\_\_

---

### Aufgabe 3 (10 Punkte)

Schreiben Sie eine Funktion `function y=mittel(N)` die den Durchschnitt (arithmetisches Mittel) der Werte einer **beliebigen** Matrix `N` berechnet und als skalaren Wert `y` zurückgibt. Die MATLAB builtin-Funktionen `mean` und `sum` dürfen **nicht** verwendet werden!

Beispiel:  $N = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow y = 3.5$

**Lösung:**

`function y=mittel(N)`

---

---

---

---

---

---

---

---

---

---



---

## LÖSUNGEN

### Aufgabe 1 (10 Punkte, 2 pro Teilaufgabe)

a)

**Lösung:** Im Wertebereich (Speicherbedarf) und in der Interpretation des Bitmusters.

b)

**Lösung:**

$y = \pi \cdot x / 2 \cdot (x < 0) + \pi / 3 \cdot x \cdot (x > 0)$

**oder:** `if x<0 y=pi*(x/2), elseif x==0 y=0, else y=x*(pi/3), end`

c)

**Lösung:**

`>> z+( sin(x) + sqrt( 3*log10(y)^2 / (2 + sqrt(z))^3 ) )`

d)

**Lösung:** (a) 0 (b) 1

e)

(e1) **Lösung:** 1 2 3 4 5

(e2) **Lösung:** 1 2 3 4 5

(e3) **Lösung:** 0 1 2 3 4 5

## Aufgabe 2 (10 Punkte, 2 pro Teilaufgabe)

a)

```
>> W(V(3))          -> Lösung: Fehler: Index nicht ganzzahlig
>> V([W(1),W(3)])  -> Lösung: 3.3  1.1
```

b)

**Lösung:** `y=sin(x.*(x.^2 + 3*x +1));`  
`y` ist eine 2 x 3 Matrix

c) `function x = f2(b)`

```
if (cos(b)- sin(b+1))*b > 0 Lösung (Markierung)
    x = 1
else
    x = 0
end
```

**Lösung (Begründung):** Klammerung stimmt nicht \_

d) `function y = f4(a)`

```
v = [a; pi];
if v(2) > round(v(1))
    y = v(1);
else
    y = pi + v(3); Lösung (Markierung)
end
```

**Lösung (Begründung):** `v(3)` gibt es nicht \_\_\_\_\_

e)

<pre>function v = f2(x,n) v = zeros(n,1); for i = 1 : round(n/2)+x     v(i) = i * x; end</pre>	<pre>(e1) &gt;&gt;f2(2,-1) -&gt;Lösung: Fehler (e2) &gt;&gt;f2(1,4)  -&gt;Lösung: [ 1                         2                         3                         0 ]</pre>
--	---

Problem (e1): **Lösung:** `zeros(-1,1)` geht nicht \_\_\_\_\_

Problem (e2): \_\_\_\_\_

---

### Aufgabe 3 (10 Punkte)

**Lösung:**

```
function y=mittel(N)
[rows,cols]=size(N);
y=0;
for m=1:rows
    for n=1:cols
        y = y + N(m,n);
    end
end
y = y/(rows*cols);
```

### Aufgabe 4 (10 Punkte)

**Lösung;**

```
function M=Aufgabe4(N)
rows=size(N,1);
h=1;
for k=rows:-1:1
    M(:,h) = N(k,:)' ;
    h=h+1;
end
```

**Es gibt natürlich viele weitere Möglichkeiten für die Funktion!**

# Matlab-Befehlsliste

a = [ 1 3 2 ; 4.5 0 -1 ] <b>dient nachfolgend als Beispielmatrix</b>	Erzeugung einer Matrix $a = \begin{bmatrix} 1 & 3 & 2 \\ 4.5 & 0 & -1 \end{bmatrix}$		
b = a(2, 3) b = a(1, [2,3])	Indizierung von Zeile 2, Spalte 3 der Matrix $a \rightarrow b = -1$ Indizierung von Zeile 1, Spalte 2 und 3 der Matrix $a \rightarrow b = 3 \ 2$		
b = a(1, 2:end) b = a(1, :)	Indizierung von Zeile 1, Spalte 2 bis Endespalte der Matrix $a \rightarrow b = 3 \ 2$ Indizierung von Zeile 1, alle Spalten der Matrix $a \rightarrow b = 1 \ 3 \ 2$		
a1 = [a, [10; 20] ] a2 = [a; [10, 20, 30] ]	Horizontale Verkettung $\rightarrow a1 = [1 \ 3 \ 2 \ 10; \ 4.5 \ 0 \ -1 \ 20]$ % 1 Spalte angefügt Vertikale Verkettung $\rightarrow a2 = [1 \ 3 \ 2; \ 4.5 \ 0 \ -1; \ 10 \ 20 \ 30]$ % 1 Zeile angefügt		
c = a * x (+ - * ^ ' ) c = a .* y (+ - .* .^ ./)	<b>Arithmetische Operatoren</b> Matrixoperationen (Addit., Subtrakt., Multipl., Matrixpotenz, Transponieren) elementweise Operationen (Addit., Subtrakt., Multipl., Potenz, Division) <b>MATRIXDIMENSION BEACHTEN !!!</b>		
< <= > >= == ~= ~= &   ~	<b>Vergleichsoperatoren</b> Logisches UND, logisches ODER, logisches NOT (Arbeiten alle <b>elementweise</b> ; Ergebnis ist elementbezogen <b>wahr 1</b> oder <b>falsch 0</b> !!!)		
[m,n] = size(a) m = length(b)	Dimension (Anzahl Zeilen und Anzahl Spalten) der Matrix $a \rightarrow m = 2 \ n = 3$ Länge des Vektors $b$ (wenn $b=[1 \ 2 \ 3]$ , $m=length(b) \rightarrow m = 3$ )		
b = ones(m,n) b = zeros(m,n)	Erzeugung einer m-mal-n Matrix $b$ mit alle Werte=1 Erzeugung einer m-mal-n Matrix $b$ mit alle Werte=0		
b = abs(a)  b = sqrt(a) b = sin(a) (cos, tan, asin, acos, ...) b = log(a), b = log10(a) b = exp(a) b = round(a)	<b>Mathematische Funktionen</b> (Arbeiten alle <b>elementweise</b> ) Absolutwerte der Matrixelemente in $a \rightarrow b = 1 \ 3 \ 2$ 4.5 0 1 Quadratwurzelwert der Matrixelemente in $a$ Sinuswerte der Matrixelemente (Bogenmaß) in $a$ Analog sin(a) $log(a)$ Natürlicher Logarithmus ..., $log10(a)$ Dekadischer Logarithmus ... $exp(a)$ Exponentialfunktion $e^a$ ... Auf- bzw. abrunden zum nächsten ganzzahligen Wert		
b = mean(a) b = sum(a) b = min(a) bzw. b = max(a)	spaltenweise Mittelwertberechnung Matrix $a \rightarrow b = 2.75 \ 1.50 \ 0.5$ spaltenweise Summation Matrix $a \rightarrow b = 5.5 \ 3.0 \ 1.0$ spaltenw. min. Wert Matrix $a \rightarrow b = 1 \ 0 \ -1$ (bzw. max. Wert)		
<b>function</b> [x,y] = f1(a,b)	<b>Funktionskopf mit:</b> 2 Rückgabeveriablen $x, y$ , Funktionsname $f1$ und 2 Eingangsvariablen $a, b$		
b = input('Gebe Matrix b ein:')	Interaktives Einlesen von Werten auf Variable $b$		
disp(a)  disp('Otto')	Bildschirmausgabe: $1.0 \ 3.0 \ 2.0$ 4.5 0 -1.0 Bildschirmausgabe: <i>Otto</i>		
plot(x,y,'optionen')	Datenausgabe in einem 2-dimensionalen Plot mit x Vekt. x-Koord., y Vekt. y-Koord. <i>optionen</i> z.B. 'b+' ... Ausgabe der Werte als blaue + Zeichen		
<b>for</b> k = sw : sr : ew Anweisungen <b>end</b> sw Startwert sr Schrittweite ew Endwert	Operatoren in <b>if, elseif</b> und <b>while</b> : < <= > >= == ~= ~= &   ~	<b>if</b> a < b, Anweisungen, <b>elseif</b> a > b, Anweisungen, <b>else</b> Anweisungen, <b>end</b>	<b>while</b> a < b Anweisungen <b>end</b>
b = load('file.dat') save('file.dat','x','y','-ascii')	ASCII-Daten-File <i>file.dat</i> in Variable $b$ laden Variable $x$ und $y$ in ASCII-Daten-File <i>file.dat</i> speichern		
int8, uint8, char, double b = uint8(a)	Elementweise Typkonvertierung der Werte einer Matrix Konvertiere Werte in $a$ in „unsigned 8 Bit Integer“ $\rightarrow b = 1 \ 3 \ 2$ 4 0 0		
P1= <b>struct</b> ('name','Klaus',... 'alter', 32, ... 'tel', [123, 1777])  b= P1.name P1.alter= 22 Pvec=[P1,P2]	Erzeugung einer Variablen $P1$ vom Typ <i>struct</i> (Struktur) mit den Elementvariablen $name$ (Typ <i>String</i> ), $alter$ (Typ <i>double size 1x1</i> ), $tel$ (Typ <i>double size 1x2</i> )  Elementvariable $name$ in Strukturvariable $P1$ lesen und in $b$ speichern $\rightarrow b = 'Klaus'$ Elementvariable $alter$ in Strukturvariable $P1$ auf 22 setzen Erzeugung eines Vektors $Pvec$ mit den Variablen $P1$ und $P2$ vom Typ <i>struct</i>		

# Zusammenfassung der Operatoren

Operator	skalare Operation	elementweise Matrix	Matrix-operation
Fkt.aufruf, Priorit.	()		
Transponierung Potenz	^ .^	.^	' ^
Negation NOT	- ~		
Multiplikation Division (nicht behandelt)	* .* / ./ \ .\	.* ./ .\	* / (bes. Bed.) \ (bes. Bed.)
Addition, Subtraktion	+ -		
Colon Operator			:
Vergleichsop.	< <= > >= == ~=		
UND, ODER	&   &&	&	
Zuweisung	=		

Operatorpräzedenz ↓