

Klausur zur BSc-Vorlesung „Rechnergestützte Modellierung“ des WS 2011 2. Termin

Die Klausur besteht aus drei Aufgaben. In der ersten Aufgabe sind 20 Punkte zu erreichen, in der zweiten Aufgabe 5 und in der letzten 10 Punkte.

Am Ende der Klausur finden Sie Platz für Nebenrechnungen. Sollte Ihnen bei der Bearbeitung von Aufgabe 3 ein Fehler unterlaufen, finden Sie am Ende der Klausur zusätzlich die Möglichkeit Ihre Lösung nochmal gut leserlich zu präsentieren. Vergewissern Sie sich bitte bei der Abgabe, dass Sie alle ungültigen Lösungsversuche durchgestrichen haben!

Füllen Sie bitte zunächst die folgende Tabelle aus. Schreiben Sie bitte anschließend Ihren Namen auf jede Seite dieser Klausur!

Vorname	
Nachname	
Matrikelnummer	

Viel Erfolg!

Die nachfolgende Tabelle ist nur für die interne Verwendung gedacht! Nehmen Sie hier keine Eintragungen vor!

Aufgabe 1	Aufgabe 2		Aufgabe 3	Summe	Note
	a	b-d			

Aufgabe 1 (20 Punkte)

Untersuchen Sie das folgende Programm auf Fehler. Tragen Sie anschließend den jeweils korrekten Befehl **vollständig** mit der zugehörigen Zeilenzahl in der nachfolgenden Tabelle ein.

Hinweis: Die Zeilenzahlen sind nicht Bestandteil des Programms und dienen nur der Übersichtlichkeit! **Es ist nicht gestattet, die Reihenfolge der Befehle zu verändern, weitere Befehle hinzuzufügen oder vorhandene Befehle vollständig zu entfernen!**

```
01 funtcion Oligopolist(sw)
02 Einlesen;
03 (qmax,exitflag,fval) = fmaxsearch(Gew,sw);
04 if exitflag==1
05     clear all;
06     fprintf('Im Gleichgewicht beträgt die Produktion des Oligopolisten');
07     fprintf(':\n\ng = 0.2f',q);
08     fprintf('\n\nund die Gewinne sind \n\n');
09     fprintf('Gew = 0.2f\n',-Gew[q]);
10 elseif
11     fprintf('Es konnte keine Lösung gefunden werden! \n');
12     fprintf('Da es sich um ein nichtlineares Gleichungssystem handelt,');
13     fprintf('\nmacht es Sinn, einen anderen Startwert zu probieren!\n');
14 end;
15 end
16
17 function G=Gew[q]
18 global globvar;
19 Erloes=N[q+globVar(4)]*q;
20 Kosten=globVar(3)*q;
21 G=Kosten-Erloes;
22 end
23
24 function P=N(q)
25 global globvar;
26 a=globVar(1);
27 Pe=a*globVar(2)/Q;
28 end
29
30 function ()=Einlesen
31 global globvar;
32 fprintf('Bitte geben Sie nacheinander die Konsumelastizität für \n');
33 fprintf('dieses Gut,das Einkommen des Konsumenten, die \n');
34 fprintf('Stückkosten des Oligopolisten sowie den Output des \n');
35 fprintf('anderen Oligopolisten ein: \n');
36 while i=1:4
37     globVar=fprintf('>');
38 end;
39 end
```


Aufgabe 2 (5 Punkte)

Nehmen Sie an, Sie hätten die folgende Funktion programmiert und in Ihrem Arbeitsverzeichnis gespeichert:

```
01 function y=test(z)
02 clc;
03 s=size(z,1);
04 r=size(z,2);
05 if min([s;r])>1
06     fprintf('Bitte andere Übergabe wählen');
07 else
08     for i=1:max([r;s])-1
09         k=max([r;s]);
10         while k>i
11             t=z(k-1);
12             z(k-1)=z(k);
13             z(k)=t;
14             k=k-1;
15         end;
16     end;
17     y=z;
18 end;
19 end
```

a) Beschreiben in maximal 3 Sätzen den Sinn des Programms. (2P)

b) Was wäre die Bildschirmausgabe, wenn man die folgenden Befehle ausführen würde: (1P)

```
y=[1;5;4;7];
x=test(y)
```

Name: _____

- Platz für Nebenrechnungen -

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to write down their calculations or work.

- Fortsetzung Aufgabe 2 -

```
01 function y=test(z)
02 clc;
03 s=size(z,1);
04 r=size(z,2);
05 if min([s;r])>1
06     fprintf('Bitte andere Übergabe wählen');
07 else
08     for i=1:max([r;s])-1
09         k=max([r;s]);
10         while k>i
11             t=z(k-1);
12             z(k-1)=z(k);
13             z(k)=t;
14             k=k-1;
15         end;
16     end;
17     y=z;
18 end;
19 end
```

c) Was wäre die Bildschirmausgabe, wenn man die folgenden Befehle ausführen würde: (1P)

```
x=[6 8 3 5];
z=test(x)
```

d) Was wäre die Bildschirmausgabe, wenn man die folgenden Befehle ausführen würde: (1P)

```
x=[1 5 4 7].';
x=test(x);
x=test(x)
```

Name: _____

- Platz für Nebenrechnungen -

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to write down their calculations or work.

Aufgabe 3 (10 Punkte)

Das folgende Programm soll die Werte des keynesianischen Multiplikators über mehrere Jahrzehnte berechnen und graphisch darstellen.

Aufgabe: Schreiben Sie eine Funktion mit zwei Über- und einer Rückgabevervariablen. Die Übergabevervariablen nennen Sie "c1" und "T". Die Rückgabevervariable soll den Namen "erg" tragen. Nutzen Sie die Variable "c1", um Ihrer Funktion den Wert der marginalen Konsumquote zu übergeben und die Variable "T" zur Übergabe der Anzahl an Jahrzehnten. Lassen Sie Ihre Funktion die Werte des keynesianischen Multiplikators errechnen und in einer Matrix speichern. Speichern Sie dabei die Größe der Multiplikatoren eines Jahrzehnts in einer eigenen Zeile. Anschließend nutzen Sie den Ihnen aus der Vorlesung bekannten Befehl zum Erstellen einer Grafik, um die jährliche Entwicklung des Multiplikators darzustellen. Betiteln Sie Ihre Grafik mit "Multiplikator". Abschließend geben Sie die von Ihnen bereits errechnete Matrix mit den Multiplikatoren zurück. Ein genaues Rechenbeispiel finden Sie auf der folgenden Seite.

Tipp: Um den Multiplikator grafisch darzustellen, empfiehlt es sich seine Werte zusätzlich zu der Variable "erg" in einer zweiten Variablen zu speichern. Diese zweite Variable sollte ein für den *plot*-Befehl direkt geeignetes Format besitzen!

Bei der Bearbeitung der Aufgabe ist es Ihnen nicht erlaubt, den vorgegeben Code zu modifizieren! Nutzen Sie ausschließlich die vorgegebenen Stellen, um eigenen Code zu ergänzen!

Wichtig: Sie müssen nicht zwingend alle möglichen Stellen zur Bearbeitung dieser Aufgabe verwenden. Vielmehr ist es durch eine effiziente Programmierung möglich, auf einige Platzhalter zu verzichten! Sollten Sie mehr Platz benötigen, können Sie selbstverständlich mehrere Befehle in eine Reihe schreiben!

Name: _____

Ein Zahlenbeispiel:

Lassen Sie uns die geforderten Berechnungen für eine marginale Konsumquote von

$$c_1 = 0.8$$

und einen Horizont von 3 Jahrzehnten durchführen. Wie Sie aus der Vorlesung wissen, können Sie den keynesianischen Multiplikator in Jahr t gemäß der Formel:

$$\text{Multiplikator}_t = \sum_{i=0}^{t-1} c_1^i$$

bestimmen. Übertragen auf das Beispiel erhalten wir:

Jahr	Größe des Multiplikators
1	$\text{Multiplikator}_{t=1} = \sum_{i=0}^0 0.8^i = 0.8^0 = 1$
2	$\text{Multiplikator}_{t=2} = \sum_{i=0}^1 0.8^i = 0.8^0 + 0.8^1 = 1.8$
3	$\text{Multiplikator}_{t=3} = \sum_{i=0}^2 0.8^i = 0.8^0 + 0.8^1 + 0.8^2 = 2.44$
---	...
30	$\text{Multiplikator}_{t=30} = \sum_{i=0}^{29} 0.8^i = 0.8^0 + 0.8^1 + \dots + 0.8^{29} \approx 4.994$

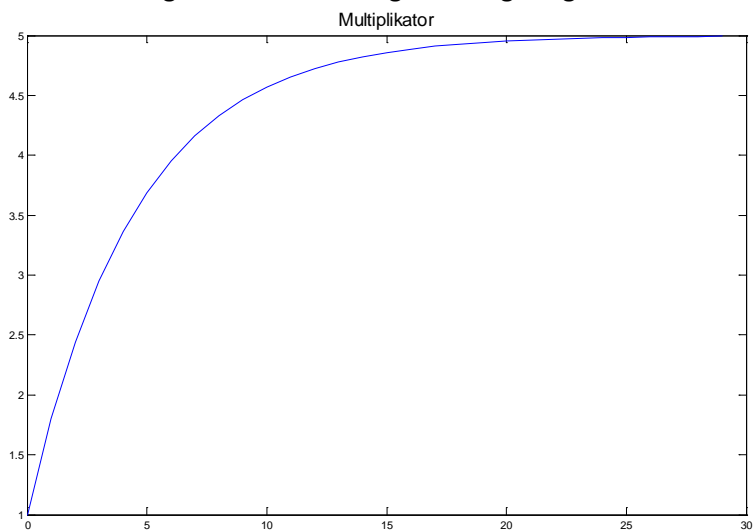
Der Aufruf `Keynes(0.8,3)` in MatLab würde daher folgende Bildschirmausgaben erzeugen:

```

Command Window
>> Keynes(0.8,3)
ans =
    1.0000    1.8000    2.4400    2.9520    3.3616    3.6893    3.9514    4.1611    4.3289    4.4631
    4.5705    4.6564    4.7251    4.7801    4.8241    4.8593    4.8874    4.9099    4.9279    4.9424
    4.9539    4.9631    4.9705    4.9764    4.9811    4.9849    4.9879    4.9903    4.9923    4.9938
fx >>

```

Zusätzlich würde automatisch folgende Grafik erzeugt und angezeigt:



Achten Sie darauf, die oben gezeigte Überschrift auch in Ihrer Funktion umzusetzen! Gehen Sie dabei davon aus, dass Sie die von MatLab automatisch gewählte Skalierung beibehalten können.

Viel Erfolg!

```
function _____ = Keynes _____
```

```
_____  
_____  
_____  
_____  
_____
```

```
for _____
```

```
    for _____
```

```
        _____  
        _____  
        _____  
        _____  
        _____  
        _____
```

```
    end;
```

```
end;
```

```
_____  
_____  
_____  
_____  
_____
```

```
end
```

Name: _____

- Platz für erneute Bearbeitung der Aufgabe 3 -

```
function _____ = Keynes _____  
_____  
_____  
_____  
_____  
_____  
  
for _____  
_____  
    for _____  
        _____  
        _____  
        _____  
        _____  
        _____  
        _____  
    end;  
    _____  
end;  
_____  
_____  
_____  
_____  
_____  
  
end
```

- Platz für erneute Bearbeitung der Aufgabe 3 -

```
function _____ = Keynes _____  
  
_____  
_____  
_____  
_____  
_____  
  
for _____  
    _____  
    for _____  
        _____  
        _____  
        _____  
        _____  
        _____  
        _____  
        _____  
    end;  
end;  
  
_____  
_____  
_____  
_____  
_____  
  
end
```

Name: _____

- Platz für Nebenrechnungen –

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to perform calculations or show their work.

- Platz für Nebenrechnungen -