Dr. Frank Schöpfer

Einführung in Matlab
Lösungen 2

Aufgabe 1:

(a)
```
function x=kettenbruch_test(n)
x=zeros(1,n);
x(1)=1;
for k=2:n
   x(k)=1+1/x(k-1);
end
end
```

Plotten

```
>> n=10; x=kettenbruch_test(n); plot(1:n,x,'o',1:n,(1+sqrt(5))/2*ones(1,n))
```

(b)
```
function x=kettenbruch(n)
x=1;
for k=2:n
   x=1+1/x;
end
end
```

Vergleich mit (a)

```
>> x=kettenbruch_test(10); x(10), x=kettenbruch(10)
ans =
    1.6182
x =
    1.6182
```

(c)
```
function x=kettenbruch_rekursiv(n)
if n==1
    x=1;
else
    x=1+1/kettenbruch_rekursiv(n-1);
end
end
```

(d)
```
function [x,iter]=kettenbruch_while(epsilon)
x=1;
iter=0;
while abs(x-1-1/x)>epsilon
   x=1+1/x;
   iter=iter+1;
end
end
```

Test

```
>> format long
>> [x,iter]=kettenbruch_while(1e-4)
x =
   1.617977528089888
iter =
    10
>> abs(x-(1+sqrt(5))/2)
ans =
     5.646066000730698e-05
```

Aufgabe 2:

```
function b=binomial(n,k)
if (k==0) || (k==n)
    b=1;
    return
end
if (k==1) || (k==n-1)
    b=n;
    return
end
b=binomial(n-1,k)+binomial(n-1,k-1);
end
```

Aufgabe 3:

```
function [x,indices]=selection_sort2(x)
n=numel(x);
indices=1:n; % für (b)
if n==1
    return
end
for j=1:n
    ind=finde_min(x,j,n);
    if ind~=j
        x=tausche(x,j,ind);
        indices=tausche(indices,j,ind); % für (b)
    end
end
end
function ind=finde_min(x,j,n)
ind=j;
for k=j+1:n
   if x(k)<x(ind)
       ind=k;
   end
end
end
function x=tausche(x,ind1,ind2)
a=x(ind1);
x(ind1)=x(ind2);
x(ind2)=a;
end
```

Hier ist die nicht-rekursive Version etwas schneller.

```
>> x=rand(1,1e4);
>> tic; xs=selection_sort(x); toc
Elapsed time is 0.726336 seconds.
>> tic; xs2=selection_sort2(x); toc
Elapsed time is 0.316230 seconds.
```