Humboldt-Universität zu Berlin
Institut für Mathematik
Dr. L. Lehmann; Dr. A. Laurain;
Dipl.Math. T. Bosse

Probeklausur zur Vorlesung
Mathematikorientierte Computernutzung (SS 11)

Aufgabe 1:

Füllen Sie den Latex-Lückentext auf der rechten Seite so aus, dass Sie den folgenden Originaltext erhalten:

Sekantenverfahren

Sei $f \in \mathcal{C}[a,b]$ mit $f(a)f(b) < 0$ und existiere mindestens eine Nullstelle (a,b). So lässt sich mittels des *Sekantenverfahrens*, eine Nullstelle $x^* \in (a,b)$ von f bestimmen. Das iterative Verfahren ist dabei wie folgt definiert:

`Startwert:` $x^{(0)}$, $f_0 := f\left(x^{(0)}\right)$ und $x^{(1)} := f\left(x^{(1)}\right)$ mit $f_0 * f_1 < 0$.

`Iterationsvorschrift:`

$$x^{(v+1)} = x^{(v)} - \frac{x^{(v)} - x^{(v-1)}}{f_v - f_{v-1}} * f_v$$

für $v = 1, 2, \ldots$ und $f_v - f_{v-1} \neq 0$.

Beispiel:

- Man berechne eine Nullstelle der Funktion $f(x) = x^3 + 2x - 6$ in dem Intervall $(1,2)$ mittels des Sekantenverfahrens und notiere die dabei entstehenden Zwischenwerte in der folgenden Tabelle:

$x^{(v)}$	f_v	$x^{(v-1)}$	f_{v-1}	$\frac{x^{(v)} - x^{(v-1)}}{f_v - f_{v-1}}$
2	6	1	-3	$0,666667$
$1,333333$	$-0,962963$	2	6	$-0,092199$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

```
% Alle entsprechenden Pakete sind bereits eingebunden
\begin{document}
(1) Sekantenverfahren (2)

Sei (3) mit (4) und existiere mindestens eine Nullstelle
(5). So l"asst sich mittels des (6), eine
Nullstelle (7) von (8) bestimmen. Das iterative Verfahren
ist dabei wie folgt definiert:(9)

(10): (11), (12)
  und (13) mit (14).(9)

(15): \begin{(16)}
        (17)
      \end{(18)}
      f"ur (19).(9)

(20):
\begin{(21)}
 (22) Man berechne eine Nullstelle der Funktion (23)
      in dem Intervall (24) mittels der Sekantenverfahrens
      und notiere die dabei entstehenden Zwischenwerte
      in der folgenden Tabelle:
      \begin{(25)}{(26)}
        (27)
      \end{(28)}
\end{(29)}
\end{document}
```

Tragen Sie bitte hier die entsprechenden Latex Befehle ein:

(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

(13)

(14)

(15)

(16)

(17)

(18)

(19)

(20)

(21)

(22)

(23)

(24)

(25)

(26)

(27)

(28)

(29)

Aufgabe 2:

1. Berechnen Sie $x^{(3)}, \ldots, x^{(6)}$ mittels des Verfahrens aus Aufgabe 1 und schreiben Sie die Werte in die folgende Tabelle:

v	$x^{(v)}$	f_v	$x^{(v-1)}$	f_{v-1}	$\frac{x^{(v)} - x^{(v-1)}}{f_v - f_{v-1}}$
1	2	6	1	-3	$0,666667$
2	$1,333333$	$-0,962963$	2	6	$-0,092199$
3					
4					
5					
6					

2. Auf der rechten Seite sehen Sie ein mögliche Implementierung des Verfahrens für $v = 1, \ldots$ bis $f(x^{(v)}) - f(x^{(v-1)}) = 0$. Leider haben sich ein paar Fehler eingeschlichen. Markieren Sie alle Fehler im Code und schreiben Sie die nötigen Korrekturen auf die nächste Seite (mit kurzer Begründung), sodass sich der Code kompilieren lässt und das richtige Ergebnis liefert.

%SekantenVerfahren.java:

```
import HUMath.Algebra.*;

public class sekantenverfahren {

 public static int f(x) {
  return Math.pow(x,3.0)+2.0x-6.0;
 }

 public static void main(String args[] ){

  B.wl(Untere Intervallgrenze:);
  double x1 = B.read()
  B.wl(Obere Intervallgrenze:);
  double x2 = B.read()
  double f1=f(x1);
  double f2=f(x2);

  {
    double xnew=x2-(x2-x1)%(f2-f1)*f2;
    double fnew=f(xnew);
    x1=x2; f1=f2;
    x2=xnew; f2=fnew;
    B.wl("x1="+x1+", f1="+f1+", x2="+x2+", f2="+f2);
  }
  while(Math.abs(f1-f2)<0.0)
```

Tragen Sie bitte hier die nötigen Verbesserungen mit der jeweils entsprechenden Begründung ein:

-
-
-
-
-
-
-
-
-
-
-
-
-
-

Aufgabe 3:

1. Untersuchen Sie den Java Code auf der rechten Seite. Beschreiben Sie seine Funktion und ermitteln Sie das Ergebnis für die Eingaben:

   4, 254, -3, 8 und 13.

   Geben Sie für obiges Beispiel alle wesentlichen Zwischenschritte an, die bei der Ausführung des Codes passieren.

2. Ersetzen Sie die '**for**'-Schleife:

```
for(int i=0;i<f.length/2;i++)
  {
    int temp = f[i];
    f[i] = f[f.length-1-i];
    f[f.length-1-i] = temp;
  }
```

   durch eine rekursive Funktion so, dass das Ergebnis unverändert bleibt. Mit anderen Worten, schreiben Sie auf dem Papier eine komplilierfähige rekursive Funktion mit dem entsprechenden Aufruf, die anstatt der Schleife ausgeführt werden soll. Dokumentieren Sie ihren Code.

```
// Blackbox.java

import HUMath.Algebra.*;

public class Blackbox {

public static void main(String args[]){
  B.wl("Eingabe 1:");
  int laenge = B.readint();
  B.wl("Eingabe 2:");
  int[] f = new int[laenge];
  for (int i=0; i<f.length; i++) {
   B.wl("f["+i+"]:");
   f[i] = B.readint();
  };

  for(int i=0;i<f.length/2;i++)
  {
    int temp = f[i];
    f[i] = f[f.length-1-i];
    f[f.length-1-i] = temp;
  }

  B.wl("Ergebnis:");
  for (int i=0;i<f.length;i++)
   B.wr(" " + f[i]);
  B.wl();
 }
}
```