

Ingenieurinformatik

Name	Vorname	Matrikelnummer	Sem.-Gr.:	Hörsaal	Platz

Zulassung geprüft
vom Aufgabensteller:

Teil 1	Aufg. 2	Aufg. 3	Aufg. 4	Summe	Note

Teil 1/Aufgabe 1: 30 Minuten ohne Unterlagen, **Teil 2/Aufgaben 2-4:** 60 Minuten, beliebige eigene Unterlagen aber keine PC/Laptops, Bearbeitung mit Bleistift erlaubt.

Die Prüfung ist nur dann gültig, wenn Sie die erforderliche Zulassungsvoraussetzung erworben haben (drei Testate im Praktikum). Dies wird vom Aufgabensteller überprüft.

Aufgabe 2: (ca. 20 Punkte)

Schreiben Sie ein C-Programm, das zwei komplexe Zahlen multipliziert oder dividiert.

Der Anwender wählt zunächst die gewünschte Aktion aus (1 = Multiplikation, 2 = Division, 0 = Beenden, Falscheingaben müssen nicht berücksichtigt werden) und gibt die Real- und Imaginärteile der beiden Zahlen ein. Nun werden Real- und Imaginärteil des Ergebnisses berechnet und mit zwei Nachkommastellen ausgegeben. Bei einer Division durch null erfolgt eine entsprechende Fehlermeldung.

Das Programm führt solange immer neue Berechnungen aus, bis der Anwender als gewünschte Aktion „0“ eingibt. In diesem Fall wird das Programm beendet.

Rechenregeln:

$$z_1 = a + ib$$

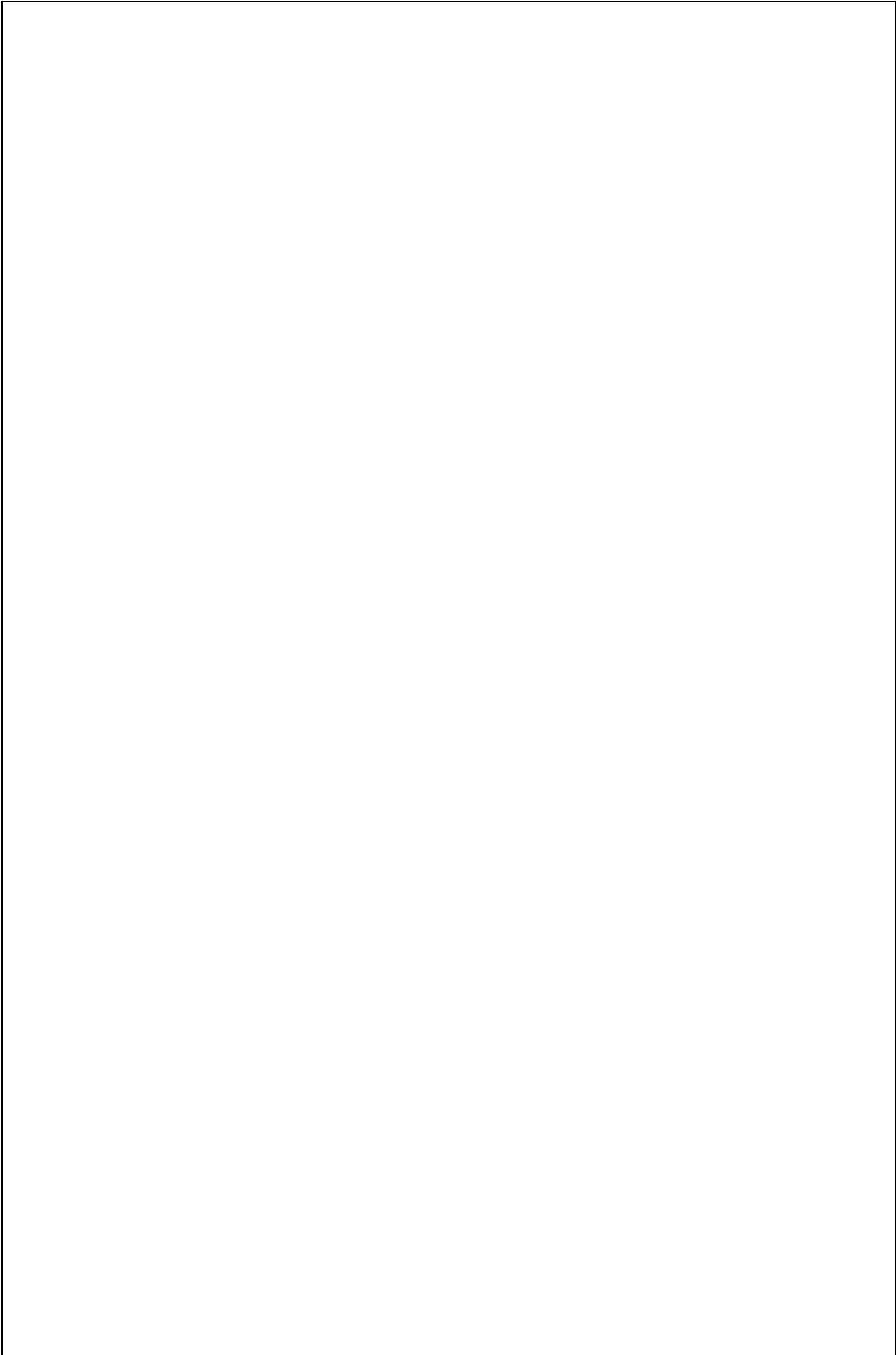
$$z_2 = c + id$$

$$z_1 \cdot z_2 = (ac - bd) + i(ad + bc)$$

$$\frac{z_1}{z_2} = \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2}$$

```

C:\Windows\system32\cmd.exe
<1> Mult., <2> Div. oder <0> Ende: 1
Zahl1, Re: 2
Zahl1, Im: 4
Zahl2, Re: 0
Zahl2, Im: 2
Ergebnis, Re = -8.00, Im = 4.00
<1> Mult., <2> Div. oder <0> Ende: 2
Zahl1, Re: 3
Zahl1, Im: 6
Zahl2, Re: 0
Zahl2, Im: 0
Division durch null!
<1> Mult., <2> Div. oder <0> Ende: 0
Drücken Sie eine beliebige Taste . . .
  
```



Aufgabe 3: (ca. 19 Punkte)

Das Polynom $p(x) = x^5 - 3x^4 - \frac{10}{3}x^3 + 10x^2 + x - 3$ besitzt 5 reelle Nullstellen, die im x-Intervall $[-2, +4]$ liegen. Folgendes C-Programm ermittelt diese 5 Nullstellen und speichert sie zunächst in einem Vektor mit dem Namen nullst. Danach werden die Nullstellen auf dem Bildschirm ausgegeben und der Kurvenverlauf in einem Chart grafisch dargestellt.

```
/* Polynom darstellen und Nullstellen suchen */
#define DELTA = 0.001
#include <stdio.h>
#include <math.h>
#include "chart.h"

double polynom(double x)

int main(void)
{
    double x, y1, y2, nullst[];
    int anz, i;
    anz = 0; x = -2.0;
    y1 = polynom(x);

    do
    {
        if(x == -2.0)
            chart_moveto(x, y1)
        else
            chart_lineto(x, y1, CHART_RED);

        y2 = y1; x += DELTA;
        y1 = polynom(x);
        if(y1 * y2 <= 0)
            nullst[anz++] = x - 0.5 * DELTA;
    }
    while(x <= 4.0);

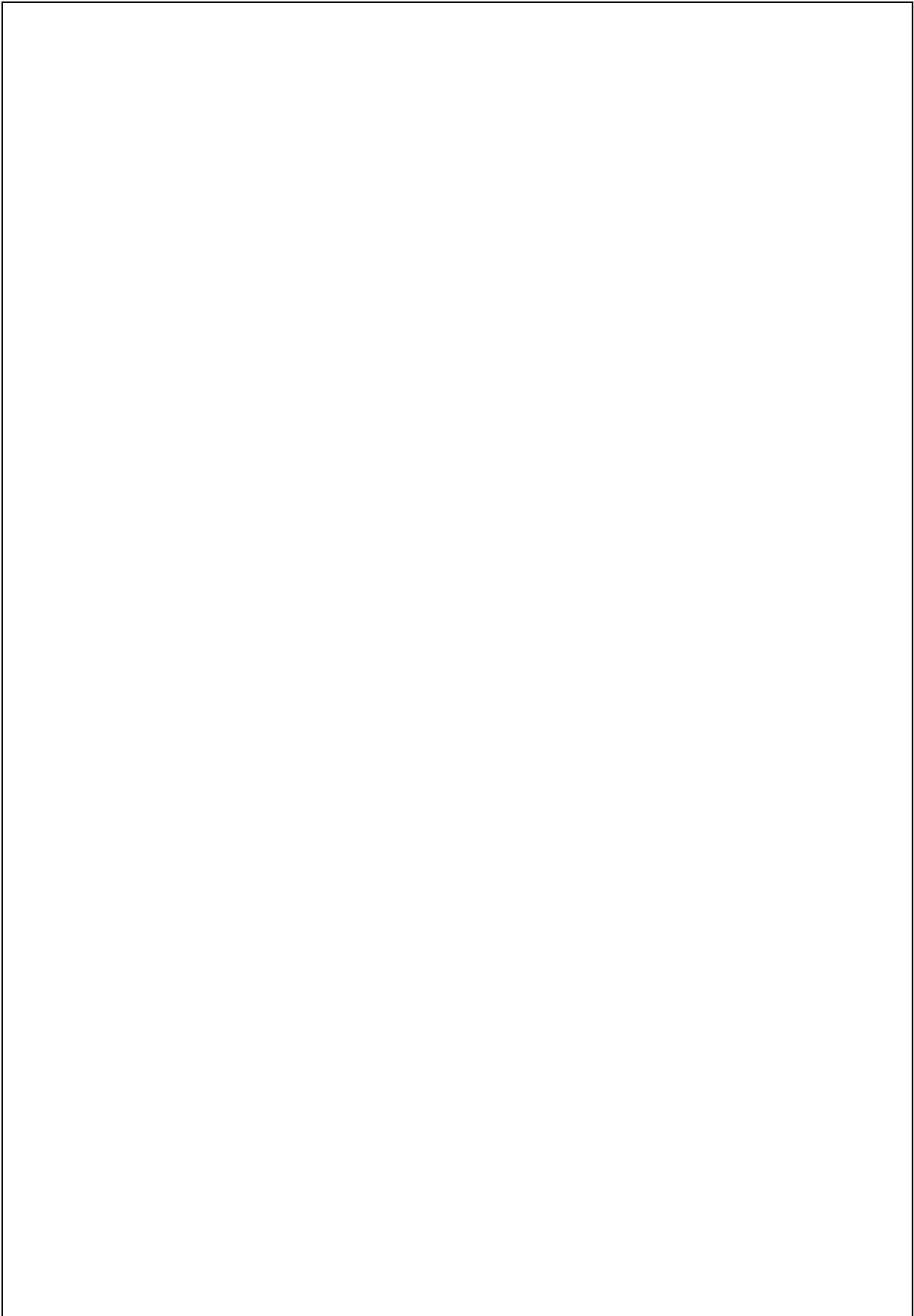
    for(i = 0; i < anz; ++i)
        printf("x%f = %.3f\n", i + 1, nullst[i]);
    printf("Aktueller Wert von i = %d", i);          /**/
    chart_show(CHART_DEFAULT, "Polynom");
    return 0;
}

double polynom(double x)
{
    return pow(x, 5.0) - 3 * pow(x, 4) -
        10.0 / 3.0 * pow(x, 3) + 10.0 * x * x + x - 3.0;
}
```

- 3.1. Korrigieren Sie die fünf Fehler, die sich in den Quelltext eingeschlichen haben.
- 3.2. Zeichnen Sie für das Hauptprogramm (Funktion main) ein Struktogramm, welches den genauen Ablauf dieser Funktion beschreibt.
- 3.3. Gehen Sie davon aus, dass das Programm fünf Nullstellen gefunden hat (Variable `anz == 5`). Kurz vor dem Ende des Hauptprogramms ist eine Zeile mit `/**/` markiert, dort wird der aktuelle Wert der Variablen `i` ausgegeben. Welcher Wert wird ausgegeben?

Aktueller Wert von `i` =

Struktogramm:



Aufgabe 4: (ca. 28 Punkte)**Name:** _____

- 4.1. In einem C-Programm sollen 10 Messwerte (Datentyp: double) eingegeben werden. Das Programm ermittelt anschließend, wie viele dieser Messwerte negativ sind und gibt deren Anzahl auf dem Bildschirm aus. Die Funktionen „eingabe“ und „negativ“ müssen noch definiert werden. Fügen Sie die beiden Funktionsdefinitionen zum C-Quelltext hinzu!

Die Funktion „main“ soll nicht verändert werden. Verwenden Sie keine globalen Variablen!

```
#define _CRT_SECURE_NO_WARNINGS
#define ANZAHL 10
#include <stdio.h>

void eingabe(double *messw);
int negativ(double *messw);

int main(void)
{
    int n;
    double x[ANZAHL];

    eingabe(x);
    n = negativ(x);
    printf("Anzahl neg. Messwerte: %d\n", n);
    return 0;
}
```

4.2. Wie lautet die Ausgabe des folgenden Programms?

```
#include <stdio.h>
int main(void)
{
    int row, col;
    double a, b, mat[3][3] =
    {
        { 1.0, 2.0, 3.0 },
        { 4.0, 5.0, 6.0 },
        { 7.0, 8.0, 9.0 }
    };

    a = 0;
    for(row = 0; row < 3; ++row)
    {
        b = 0;
        for(col = 0; col < 3; ++col)
        {
            b += mat[row][col];
            a += mat[row][col];
        }
        printf("%.0f\n", b);
    }
    printf("%.0f\n", a);
    return 0;
}
```

Ausgabe:

4.3. Gegeben ist das folgende C-Programm zur Verarbeitung einer Zeichenkette:

```
/* Arbeiten mit Zeichenketten */
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
    int i;
    char ch, text[100];
    printf("Text, max. 99 Zeichen: ");

    for(i = 0; text[i] != 0; ++i)
    {
        ch = text[i] - 1;
        printf("%c\n", ch);
    }
    printf("\ni = %d\n", i);
    return 0;
}
```

Ausgabe:

4.3.1. Fügen Sie an der markierten Stelle einen scanf-Funktionsaufruf zur Eingabe der Zeichenkette „text“ ein. Achten Sie darauf, dass der Anwender nicht mehr als 99 Zeichen eingeben kann.

4.3.2. Wie lautet die Ausgabe des Programms, wenn der Anwender die folgende Zeichenkette eingibt:

Hfifjn