

<p>4. Hausaufgabe – Aufgabenstellung</p> <p>Einführung in die Informationstechnik für Ingenieure</p> <p>Sommersemester 2016</p>	Name:
	Tutoriumstermin:

Aufgabenstellung:

Bei dieser Aufgabe sollen Sie zuerst die Struktogramme zeichnen. Anschließend soll das Programm nach der Rückgabe der korrigierten Struktogramme in der von Ihnen gewählten Programmiersprache implementiert werden. Die Programmübersetzung erfolgt mittels Makefile.

Bestandteile:

- Teil I: Struktogramme
- Teil II: Quelltext und Makefile mit `clean`-Regel

Abgabetermin:

- Teil I: **in den Tutorien** vom 06.06. bis 10.06.2016
- Teil II: **in den Tutorien** vom 04.07. bis 08.07.2016

Erreichbare Punkte: 33

Anforderungen:

Es ist beim Programmieren sehr wichtig den Quellcode zu kommentieren, damit Außenstehende ihn verstehen können und man selbst auch nach einiger Zeit noch weiß, warum man was und wie programmiert hat. Wir verlangen daher von Ihnen, dass Sie Ihre Programme mit Kommentaren versehen! Andernfalls erfolgt Punktabzug. Treten beim Übersetzen Ihrer Programme Compiler- oder Syntaxfehler auf, werden diese mit **Null** Punkten bewertet.

Kopieren Sie unbedingt alle erforderlichen Dateien in Ihr Abgabeverzeichnis: `~/Hausaufgaben/Aufgabe_4/`. Achten Sie bei der Rechtevergabe darauf, dass Ihr Tutor (als **others**) auf alle relevanten Dateien zugreifen kann.

Vorbereitung:

- Struktogramme
Lesen Sie zuerst die gesamte Aufgabenstellung durch, bevor Sie mit der Entwicklung der Struktogramme beginnen. Probieren Sie außerdem das zur Verfügung gestellte Beispielpogramm aus, um sich mit der Funktionsweise des Programms vertraut zu machen. Ein funktionierendes Beispiel zur Demonstration sowie alle benötigten Dateien finden Sie unter `/home/edv1/Dateien/aufgabe04/`. Die Bildschirmausgaben Ihres Programms müssen nicht mit denen des Beispielpogramms übereinstimmen. Sie dürfen gegebenenfalls kürzen.
- Implementierung
Erstellen Sie in Ihrem Heimatverzeichnis das Abgabeverzeichnis:
`~/Hausaufgaben/Aufgabe_4/`
Kopieren Sie (falls nötig) vorgegebene Dateien in Ihr Abgabeverzeichnis und legen Sie ein Makefile zur Übersetzung Ihres Programmes an.
Das Programm soll aus verschiedenen Modulen zusammengesetzt sein, die alle in **separaten** Dateien abzulegen sind. Implementieren Sie die unten aufgeführten Module anhand Ihrer vom Tutor korrigierten Struktogramme in Fortran95 bzw. ANSI-C.

Programmbeschreibung:

In dieser Hausaufgabe soll ein Polynomrechner entwickelt werden. Der Benutzer kann ein Polynom eingeben und im Anschluss daran dessen Ableitung und/oder Stammfunktion (ohne Integrationskonstante) berechnen. Diese und das Polynom werden schließlich in einem frei wählbaren Definitionsbereich geplottet.

Das Polynom darf höchstens 5. Grades in der Form

$$p(x) = k_1 \cdot x^5 + k_2 \cdot x^4 + k_3 \cdot x^3 + k_4 \cdot x^2 + k_5 \cdot x + k_6$$

sein und seine Koeffizienten werden in einem Feld k mit 6 Einträgen gespeichert.

• Hauptprogramm *main* (`main.c/f`)

Das Hauptprogramm stellt den Rahmen des Programms dar. Hieraus werden die Unterprogramme aufgerufen, die für den Programmablauf notwendig sind.

- Das Programm gibt beim Start eine Begrüßungsmeldung an den Benutzer aus.
- Um unvorhersehbare Ergebnisse bei den Berechnungen zu vermeiden, werden zunächst alle Einträge des Koeffizienten-Feldes k mit 0 belegt.
- Nun wird das Menü ausgegeben, welches folgende Unterpunkte enthält:
 1. Polynom-Koeffizienten eingeben
 2. Stammfunktion berechnen und anzeigen
 3. Ableitung berechnen und anzeigen
 4. Programm beenden
- Der Benutzer wird aufgefordert, eine Wahl zu treffen. Diese wird in der Variablen *wahl* abgelegt. Bei falschen Eingaben wird nach der Ausgabe einer Fehlermeldung zur Neueingabe aufgefordert.
- Wählt der Benutzer den ersten Menüpunkt, wird ihm Folgendes mitgeteilt:
 1. Der Polynomgrad darf höchstens 5 sein.
 2. Die Form des Polynoms ist $p(x) = k_1 \cdot x^5 + k_2 \cdot x^4 + k_3 \cdot x^3 + k_4 \cdot x^2 + k_5 \cdot x + k_6$.
 3. Führende Nullen müssen mit angegeben werden, z.B.
 $p(x) = 2x + 5 = 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 2 \cdot x^1 + 5 \cdot x^0$.
 Dann wird der Benutzer aufgefordert, die Polynom-Koeffizienten nacheinander einzugeben. Diese werden im Feld k abgelegt. Anschließend wird das Polynom $p(x)$ auf dem Bildschirm angezeigt.
- Wählt der Benutzer den zweiten oder dritten Menüpunkt, wird zuerst das Unterprogramm **berechnung** mit den Parametern *wahl* und k und dann das Unterprogramm **grafik** aufgerufen.
- Nach Abarbeitung der Menüpunkte 1, 2 oder 3 kehrt das Programm wieder zum Hauptmenü zurück.
- Wählt der Benutzer den vierten Menüpunkt, soll das Programm nach einer Abschiedsmeldung beendet werden.

- **Unterprogramm *berechnung*** (*berechnung.c/f*)

Hier wird die Stammfunktion $P(x)$ bzw. die Ableitung $p'(x)$ berechnet.

- Zuerst wird das Polynom $p(x)$ auf dem Bildschirm ausgegeben.
- Hat der Benutzer den zweiten Menüpunkt ausgewählt, wird die Stammfunktion $P(x)$, andernfalls die Ableitung $p'(x)$ auf dem Bildschirm ausgegeben.

Hinweis: Verwenden Sie die Integrationsregel für Polynome

$$\int (k \cdot x^n) dx = \frac{k}{n+1} \cdot x^{n+1} + C$$

bzw. die Ableitungsregel für Polynome

$$\frac{d}{dx} (k \cdot x^n) = kn \cdot x^{n-1}.$$

Die „neuen“ Koeffizienten müssen nicht neuen Variablen zugewiesen werden. Als Integrationskonstante soll $C = 0$ verwendet werden.

- Der Benutzer wird aufgefordert den Definitionsbereich anzugeben, in dem die Funktionen geplottet werden sollen. Zuerst wird die linke Intervallgrenze *von* und dann die rechte Intervallgrenze *bis* eingelesen.
- Um die beiden Funktionen zu plotten, wird das eingegebene Intervall in 50 gleich breite Abschnitte geteilt. Dazu wird zunächst die Schrittweite *sw* zwischen den Auswertungsstellen x bestimmt:

$$sw = \frac{bis - von}{50}$$

- Dann werden folgende Schritte 51 mal hintereinander ausgeführt:
 1. Berechnung der Auswertungsstelle x .
Hinweis: Die erste Auswertungsstelle ist *von*, die letzte (51.) ist *bis*.
 2. Berechnung des Polynomwertes *pol* an der Auswertungsstelle x .
 3. Berechnung des Wertes der Stammfunktion *stm* oder der Ableitung *abl* an der Auswertungsstelle x .
 4. Schreiben von x , *pol* und *stm* oder *abl* in die Datei *werte.dat*.
Hinweis: In der Datei *werte.dat* stehen dann drei Spalten mit jeweils 51 Einträgen. Die dritte Spalte enthält die Werte der Stammfunktion oder der Ableitung, je nach Benutzerwahl.

Hinweis: Für x , *pol*, *stm* und *abl* sind keine Felder notwendig. Die Variablen können überschrieben werden, sobald der aktuelle Wert in der Datei abgelegt wurde.

Implementierung: Um in C potenzieren zu können, müssen Sie den Befehl `pow(basis, exponent)` aus der `libmath.so` verwenden. Binden Sie diese mit `#include<math.h>` ein und berücksichtigen Sie sie mit Hilfe von `-lm` beim Kompilieren der Objekt-Dateien (`gcc -lm -o polyrechner ...`).

In Fortran 95 potenzieren Sie so: `basis**exponent`, z.B. `x**5`.

- **Unterprogramm *grafik*** (*grafik.c/f*)

Hier werden die beiden Funktionen geplottet.

- Der Benutzer wird gefragt, ob der Plot auf dem Bildschirm angezeigt oder in einer *.ps*-Datei gespeichert werden soll. Bei falschen Eingaben wird nach einer Fehlermeldung zur Neueingabe aufgefordert.
- Das Unterprogramm ruft Gnuplot mit der entsprechenden Batchdatei *bildschirm.gnu* oder *datei.gnu* auf. *Hinweis:* Gnuplot ist weder eine Funktion, noch ein Unterprogramm. Der Aufruf von Gnuplot ist dementsprechend als gewöhnliche Struktogrammanweisung zu gestalten.
- Wählt der Benutzer das Speichern der Grafik in einer Datei, wird ihm nach dem Speichern mitgeteilt, dass die Datei unter dem Namen *grafik.ps* gespeichert wurde.

- **GnuPlot** (*datei.gnu* und *bildschirm.gnu*)

Hinweis: Beim Zeichnen der Struktogramme können Sie diesen Absatz ignorieren. Er ist nur für die grafische Darstellung im Programm relevant.

Das Plotten der Funktionen wird bei der Implementierung des Programms mit GnuPlot verwirklicht. Dazu schreiben Sie eine Batchdatei, die aus dem Programm heraus aufgerufen wird.

Ihr Plot soll dabei mindestens den folgenden Anforderungen genügen:

- Es gibt eine sinnvolle Überschrift.
- Die Achsen sind sinnvoll beschriftet.
- Es gibt eine Legende zur Unterscheidung der Graphen.
- Die Grafik ist farbig.
- Wird die Ausgabe in die Datei gewählt, soll der Graph nicht auf dem Bildschirm angezeigt, sondern ausschließlich als Postscript-Dokument abgelegt werden.

Die Batchdatei können Sie mit `gnuplot grafik.gnu` auch in der Kommandozeile testen.

Binden Sie den Aufruf der Batchdatei wie folgt in Ihr Programm ein:

```
F95  → CALL SYSTEM("...")
C    → system("...");
```