Berufskolleg Olsberg  
Paul-Oventrop-Str. 7  
D-59939 Olsberg   
www.berufskolleg-olsberg.de

Staatlich geprüfte(r)  
Technische(r) Assistent/in Informationstechnik

Laborbericht

Fach: Programmieren

Thema: Polynome

Laborbericht Nr.: 1

Name: Henrik Lammert

Klasse / Gruppe: IF2A / B

Mitarbeiter: -

Abgabedatum: Olsberg, 21.02.2017

Fachlehrer: Frau Hohmann

Bewertung:

Inhaltsverzeichnis

1 Theoretische Vorbetrachtung 1

1.1 Polynom 1

1.2 Ableitung 1

1.3 Überlegung zur Realisierung einer Funktion zur Addition 2

2 Aufgabenstellung 3

3 Durchführung 3

3.1 Aufgabe 1 3

3.2 Aufgabe 2 3

3.3 Aufgabe 3 5

3.4 Aufgabe 4 7

3.5 Aufgabe 5 8

Literaturverzeichnis 11

Abbildungsverzeichnis

Abbildung 1: eingabe() 5

Abbildung 2: ausgabe(l\_listtype \*lList) 6

Abbildung 3: ableiten(l\_listtype \*lList) 7

Abbildung 4: addition(l\_listtype \*lList)\_1 8

Abbildung 5: addition(l\_listtype \*lList)\_2 8

Abbildung 6: subtrahieren(l\_listtype \*lList)\_1 9

Abbildung 7: subtrahieren(l\_listtype \*lList)\_2 9

Tabellenverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# Theoretische Vorbetrachtung

## Polynom

Als Polynom bezeichnet man die Summe von Termen, die jeweils Produkte einer Zahl mit einer Potenz xn sind. Die allgemeine Form von Polynomen sieht folgender maßen aus:



Eine solche Funktion könnte so aussehen:

Jedes Glied eines Polynoms hat einen Koeffizient (Vorfaktor), eine Variable (meistens als *x* bezeichnet) und einen Exponenten (Hochzahl). Alle drei sind in jedem Glied vorhanden, doch können teilweise auch in der verkürzten Schreibweise weggelassen werden. So würde das obige Beispiel folgender Maßen aussehen:

f(x) = 12x4 + x2 - 5

f(x) = 12x4 + x2 + (-5)x0

## Ableitung

Ableitungen sind sehr nützlich im Bereich der Kurvendiskussion. So entspricht die 1. Ableitung, f‘(x0), geometrisch der Tangentensteigung an der Stelle x0 einer Funktion. Um eine Funktion abzuleiten, gelten bestimmte Regeln.

Faktorenregel: Konstante Faktoren bleiben erhalten:

f(x) = a\*g(x)

* f ‘(x) = a \*g‘(x)

Konstantenregel: Der konstante Summand wird beim Ableiten zu null:

f(x) = g(x) + c

* f ‘(x) = g‘(x)

Summenregel: Die Ableitung einer Summe von Funktionen ist die Summe der Ableitungen dieser Funktion:

f(x) = g(x) + h(x)

* f ‘(x) = g‘(x) + h‘(x)

Potenzregel: „Alte“ Potenz als Faktor vor *x* setzen. „Neue“ Potenz = „Alte“ Potenz minus 1.

f(x) = xn

* f ‘(x) = n\*xn-1

## Überlegung zur Realisierung einer Funktion zur Addition

Um zwei Polynome zu addieren, müssen zunächst einmal zwei Polynome vom Benutzer eingegeben sein. Dies kann entweder innerhalb der Funktion geschehen oder es können schon bestehende Polynome (Lineare Listen) an die Funktion übergeben werden. Außerdem ist eine dritte Lineare Liste für das Ergebnis notwendig.

Wenn beide Polynome vorhanden sind, sollen später immer nur zwei Glieder mit einander addiert werden. Dafür muss aber die Voraussetzung erfüllt sein, dass die Exponenten und die Variablen mit einander identisch sind. Ist das der Fall, wird zunächst ein neues Element in der Ergebnis-Liste erstellt. Dann wird die Variable und der Exponent in die Ergebnis-Liste geschrieben. Beide Koeffizienten werden dann noch mit einander addiert und ebenfalls in die Ergebnis-Liste geschrieben. Falls die Variablen nicht identisch sind, werden beide Summanden in die neue Liste gespeichert. Immer nachdem ein Summand vollständig gespeichert wurde, muss der ‚actual‘- Pointer auf das nächste Element gesetzt werden. Das geschieht solange bis das Ende von beiden Listen erreicht wurde.

# Aufgabenstellung

Polynome lassen sich gut durch lineare Listen speichern. Man fügt nur die Glieder der Polynome als Elemente der Liste zu, die auch wirklich besetzt sind.

1. Entwickeln Sie zu diesem Problem den passenden Typ **l\_elementtype**.
2. Passen Sie die Funktion „void l\_setcontent(l\_elementtype \*newelement);“ an.
3. Schreiben Sie ein Programm, welches ein Polynom vom Benutzer einliest und auf dem Monitor ausgibt.
4. Entwickeln und realisieren Sie eine Funktion, die ein Polynom ableitet(differenziert).
5. Entwickeln und realisieren Sie eine Funktion, die zwei Polynome addiert/subtrahiert.

# Durchführung

## Aufgabe 1

Der passende l\_elementtype Typ ist:

typedef struct

{

unsigned char cVar;

int iExpo;

int iKoeff;

}l\_elementtype;

Mit ihm kann man die Variable, den Exponenten und den Koeffizienten abspeichern.

## Aufgabe 2

Die Funktion l\_setcontent(l\_listtype \*lList) für das Problem würde so aussehen:

void l\_setcontent(l\_listtype \*lList )

{

// vom Benutzer einzugeben:

printf("Variable: ");

scanf(" %c",&lList->actual->ltcontent.cVar);

printf("Exponent: ");

scanf("%d",&lList->actual->ltcontent.iExpo);

printf("Koeffizient:");

scanf("%d",&lList->actual->ltcontent.iKoeff);

}

Die Lösungen zu Aufgabe 1 & 2 wurden von der Lehrerin gestellt.

## Aufgabe 3

Die Eingabe und Ausgabe vom Polynom geschieht in meinem Programm getrennt voneinander. Es sind beides eigenständige Funktionen, die zu einem bestimmten Zeitpunkt aufgerufen werden.

Der Code für die Eingabe sieht so aus:

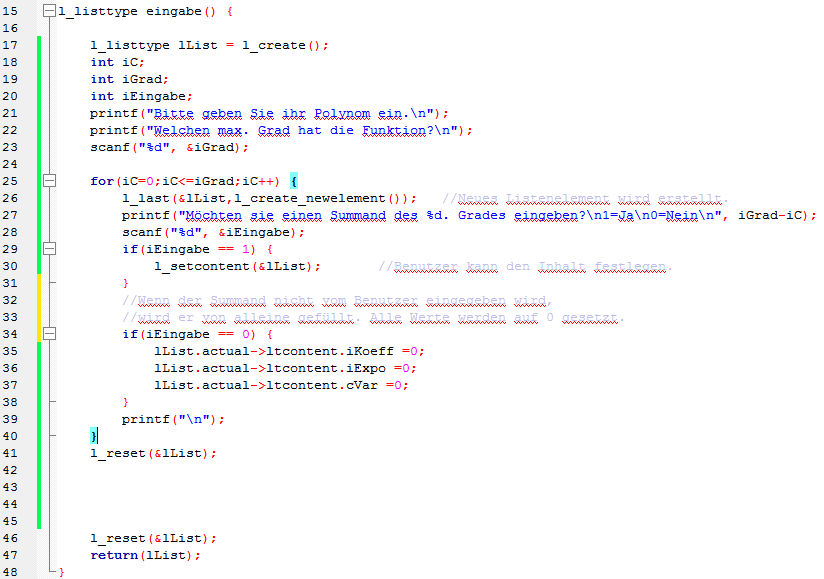


Abbildung : eingabe()

Wenn die Funktion aufgerufen wird, wird der Benutzer zunächst gefragt, welchen Grad die Funktion hat. Danach bekommt der Benutzer die Möglichkeit für jeden Grad den Summanden einzugeben. Gibt er keinen Summanden für einen bestimmten Grad ein, wird er automatisch vom Computer gefüllt. Alle Werte werden dann auf 0 gesetzt.

In der Ausgabe werden schon Regeln befolgt, um die Ausgabe zu verkürzen:

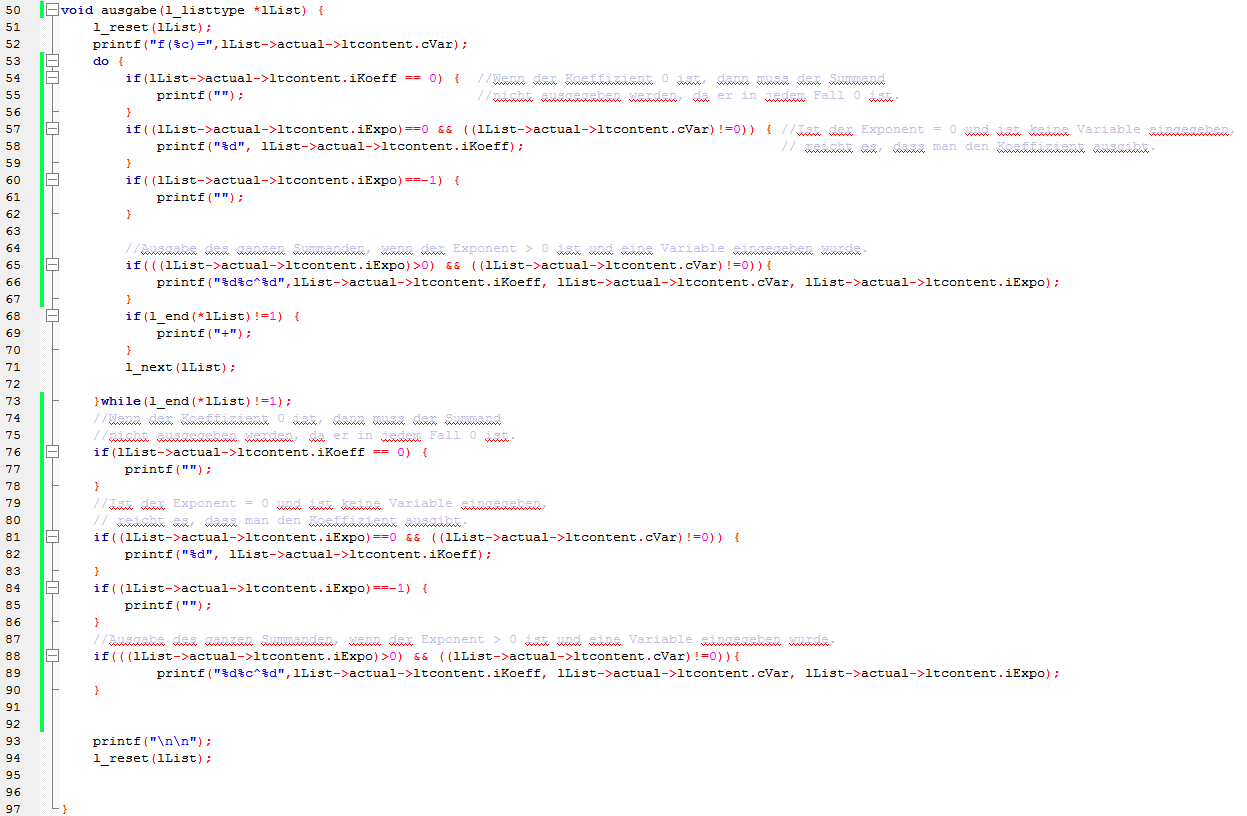


Abbildung : ausgabe(l\_listtype \*lList)

## Aufgabe 4

Die Ableitung wird mit der Potenz – und Konstantenregel realisiert. Der Abgeleitete Summand wird dann in eine neue Liste gespeichert:

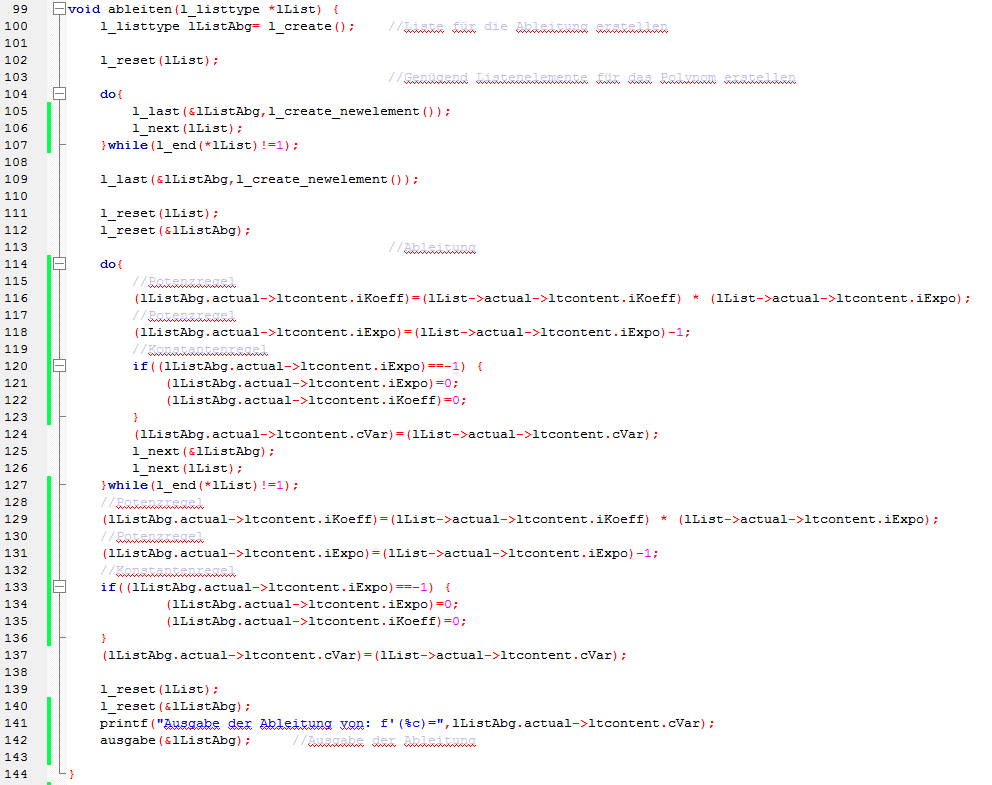


Abbildung : ableiten(l\_listtype \*lList)

## Aufgabe 5

Die Funktionen zur Addition und Subtraktion sind weitestgehend gleich. An beide Funktionen wird eine schon bestehende Liste mit einem Polynom übergeben. Die zweite Liste, mit dem das Polynom addiert/subtrahiert werden soll, wird in den Funktionen selbst erstellt. Dies geschieht, durch den Funktionsaufruf von eingabe().

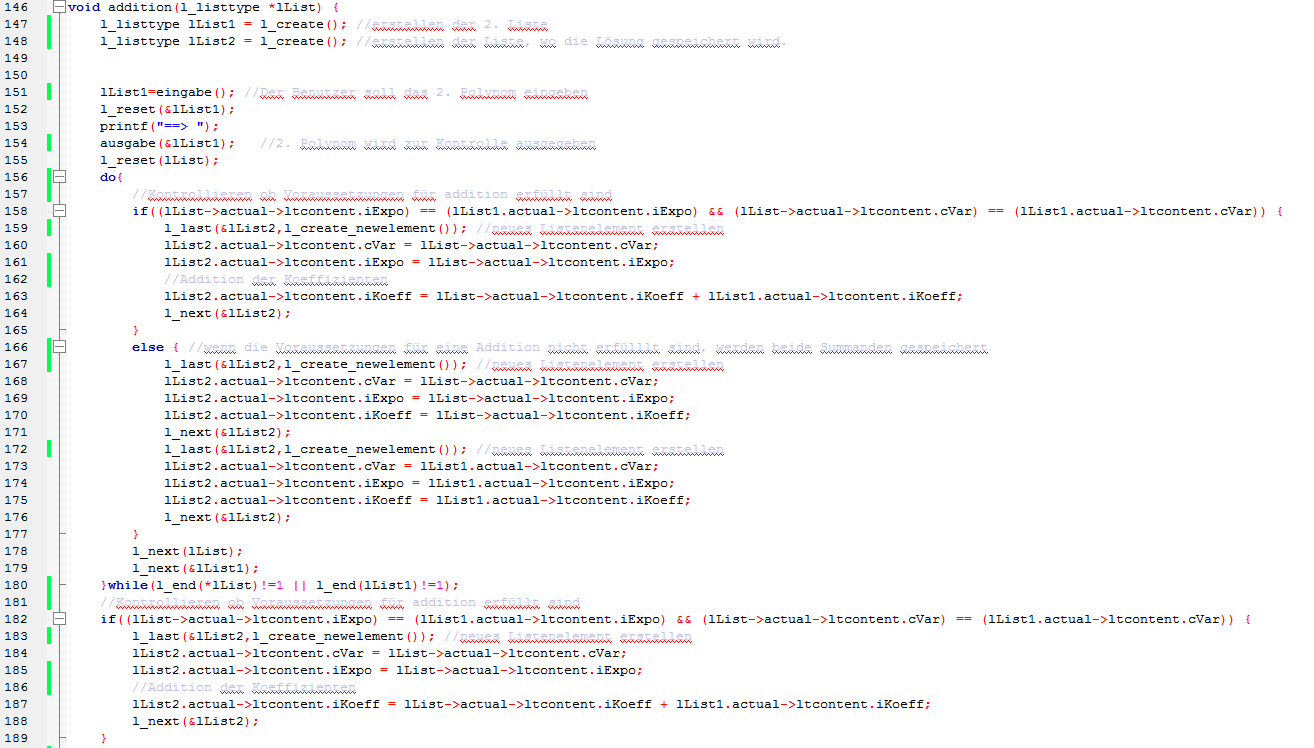
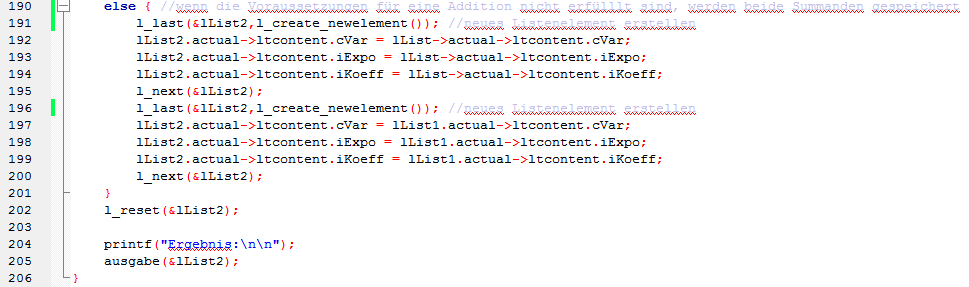
**Addition**:

Abbildung : addition(l\_listtype \*lList)\_1

Abbildung : addition(l\_listtype \*lList)\_2

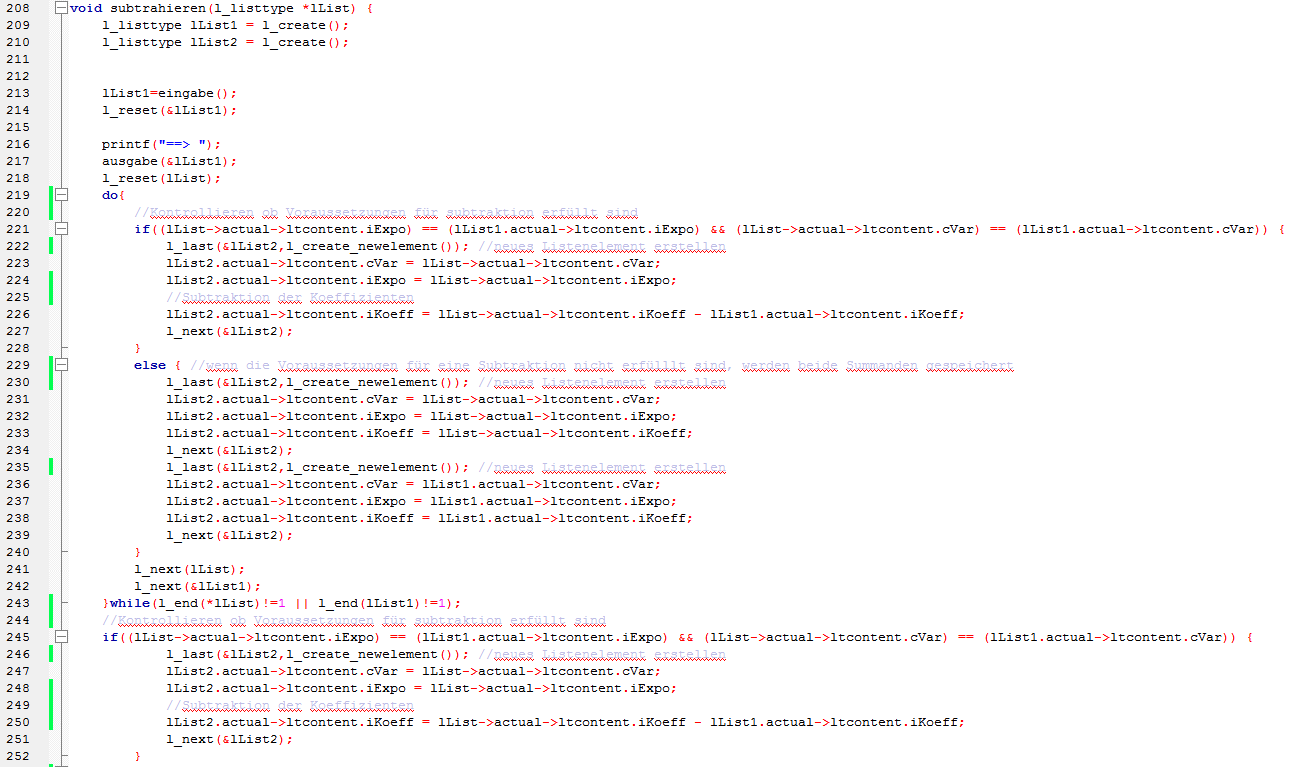
**Subtraktion**:

Abbildung : subtrahieren(l\_listtype \*lList)\_1

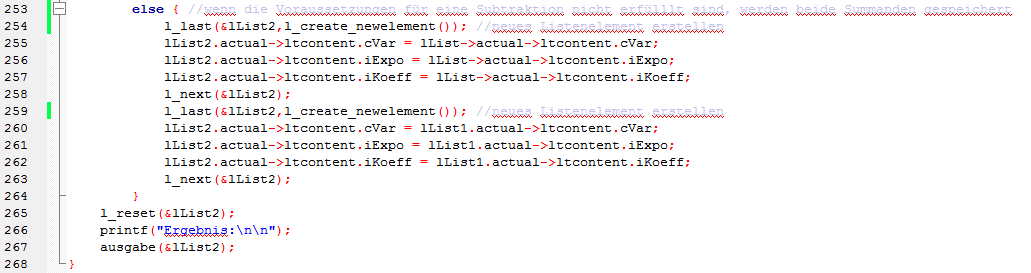
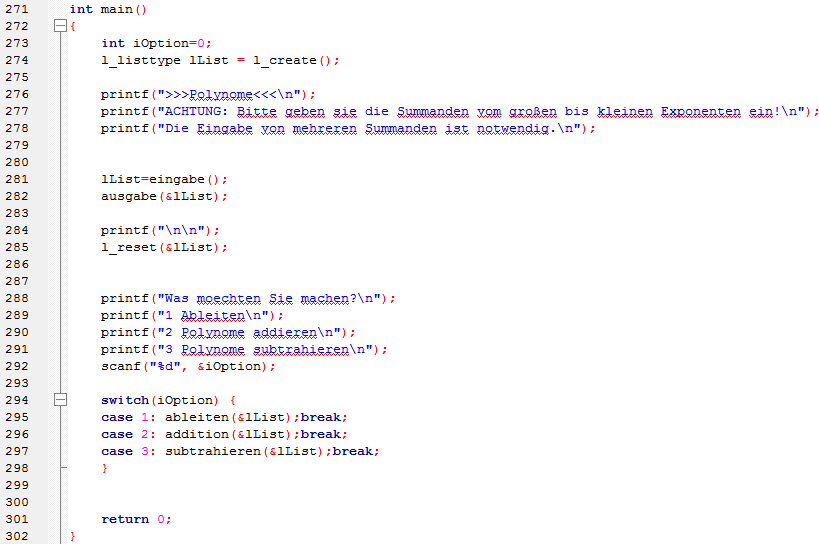


Abbildung : subtrahieren(l\_listtype \*lList)\_2

Die main()-Funktion für das Programm sieht so aus:

Es wird zunächst eine Lineare Liste für das Polynom erstellt. Daraufhin wird mit dem Funktionsaufruf eingabe(), die Liste gefüllt. Der Benutzer kann sich dann eine Option aussuchen, die er machen möchte.

# Literaturverzeichnis

[1] [bitte ersetzen, ggf. löschen]