

Studiengang Maschinenbau und Verfahrenstechnik

1. Vordiplom, Informatik I

Frühling 2001

Freitag, 9. März 2001

Name: _____

Vorname: _____

Legi-Nummer: _____

Unterschrift: _____

Aufgabe	Maximale Punktzahl	Erreichte Punktzahl	Visum
1	24		
2	25		
3	28		
4	22		
5	25		
Total	124		
Note			

Allgemeine Hinweise

- Zugelassene Hilfsmittel: **keine**.
- Legen Sie zu Anfang der Prüfung Ihre Legi neben sich auf den Tisch.
- Die Prüfung besteht aus 5 Aufgaben. Kontrollieren Sie, ob Sie alle Aufgaben erhalten haben.
- Verwenden Sie für jede Aufgabe ein separates Blatt.
- Schreiben Sie auf jedes Blatt Ihre Legi-Nummer (und nur diese!).
- Schreiben Sie deutlich, und benützen Sie keinen Bleistift.
- Pro Aufgabe darf höchstens ein gültiger Lösungsversuch abgegeben werden. Ungültige Lösungsversuche müssen klar durchgestrichen sein.
- In jeder Aufgabenstellung ist eine Lösungsstruktur angegeben. Nichtbeachten der Lösungsstruktur führt zu Punktabzug.
- Es empfiehlt sich *unbedingt*, zuerst alle Fragen durchzulesen.
- Abzugeben sind das vollständig ausgefüllte Deckblatt, die Aufgabenblätter und Ihre Lösungen.

Aufgabe 1: Fehlermeldungen, Typen & Konstanten, Kontrollstrukturen (24 Punkte)

- a) In der folgenden Aufgabe testen wir anhand einer typischen Situation beim Programmieren (wie Sie sie oft genug bei der Lösung Ihrer Übungen erlebt haben) Ihr Vermögen, Fehler zu finden, zu identifizieren und zu verbessern. Unterstützt werden Sie dabei durch Informationen, die Ihnen der Computer gibt.

Worum geht's? Wir präsentieren Ihnen vier Programmstücke, in denen jeweils ein Fehler enthalten ist. Dieser Fehler tritt entweder beim Kompilieren oder bei der Ausführung auf. Wir geben Ihnen die Fehlermeldung des Compilers oder die Fehlermeldung bei der Ausführung an.

Aufgabe Kennzeichnen Sie in den folgenden Programmen, wo der Fehler genau auftritt, beschreiben Sie, was der Fehler ist, und geben Sie einen Verbesserungsvorschlag an, der das Programm *auf irgendeine Weise fehlerlos* macht.

Lösungsstruktur Ihre Lösung enthält (1) die tatsächliche Zeilenangabe des Fehlers, (2) eine Beschreibung des Fehlers in Ihren eigenen Worten, (3) den Verbesserungsvorschlag.

Masstab Bewertet werden je Unteraufgabe die korrekte Angabe der Zeile (0.5 Punkte), die korrekte Beschreibung des Fehlers (0.5 Punkte), ein korrekter Verbesserungsvorschlag (1 Punkt). Jede Unteraufgabe gibt 2 Punkte. Die gesamte Teilaufgabe ist 8 Punkte wert.

```
i) 1  #include <iostream.h>
    2  int main()
    3  {
    4      int i,j;
    5      for (i= 1; i<=10; i++) {
    6          j= i*(i-1);
    7          if (j%2==0) { cout << i << endl;
    8      }
    9  }
```

```
ii) 1  #include <iostream.h>
     2  int main()
     3  {
     4      int i,j;
     5      for (i= 1; i<=10; i++) {
     6          j= i*(i-1);
     7          if (j%0==0) { cout << i << endl;
     8      }
     9  }
```

Fehler beim Kompilieren

In function 'int main()':
10: parse error at end of input

Laufzeitfehler

Floating exception

```
iii) 1  #include <iostream.h>
      2  struct Datum {
      3      int t;          // Tag
      4      char m[4];     // Monat
      5      int j;         // Jahr
      6  };
      7  int main()
      8  {
      9      Datum p;
     10      p.t= 9;
     11      p.m(0)= 'M'; p.m[1]='a';
     12      p.m[2]= 'r'; p.m[3]= 0;
     13      p.j= 2001;
     14  }
```

```
iv) 1  #include <iostream.h>
     2  struct Datum {
     3      int t;          // Tag
     4      char m[4];     // Monat
     5      int j;         // Jahr
     6  };
     7  int main()
     8  {
     9      Datum *p= 0;
    10      p->t= 9;
    11      p->m[0]= 'M'; p->m[1]='a';
    12      p->m[2]= 'r'; p->m[3]= 0;
    13      p->j= 2001;
    14  }
```

Fehler beim Kompilieren

In function 'int main()':
11: no matching function for
call to 'Datum::m (int)'

Laufzeitfehler

Segmentation fault

⇒

- b) In der Vorlesung haben Sie gelernt, wie man Variablen, Funktionen und Prozeduren deklariert. In dieser Aufgabe sollen Sie diese Fähigkeit demonstrieren.

Aufgabe Wir präsentieren Ihnen zwei korrekt codierte C++-Programmstücke (i-ii), aus denen klar hervorgeht, welchen Typ die „Namen“ a, b und c haben. Sie sollen für jedes Programmstück diese Programmelemente deklarieren.

Lösungsstruktur Wir erwarten uns pro Programmteil je eine Deklaration für *alle* a, b und c in korrekter C++-Syntax.

Beispiel

```
1 a= 3;
2 b= '2';
3 c= (char(a)=='2');
```

Dieser Programmteil verlangt folgende Deklaration der Objekte a, b, und c:

```
1 int a;
2 char b;
3 bool c;
```

Masstab Bewertet werden nur *notwendige* Deklarationen. Korrekte einfache Datentypen erhalten 1 Punkt. Korrekte erweiterte Datentypen werden mit bis zu 3 Punkten bewertet. Die gesamte Teilaufgabe ist 9 Punkte wert.

Die Programmstücke

i)

```
1 a[int(c)]= 'c';
2 b= 2.0;
3 c= 'a';
```

ii)

```
1 a.a= b=='@';
2 a.c= (c%int(b))*2.26;
3 b= '@';
```

- c) Die folgende Aufgabe prüft, ob Sie Schleifentypen in C++ verstanden haben.

Betrachten Sie folgendes Programmstück:

```
1 int i,b;
2 double a= 2.0;
3
4 b= ...;
5
6 for (i= 0; i<b; i++) {
7     a= a*a;
8 }
```

Dieses Programm enthält eine for-Schleife. Der Wert der Variablen b ist beliebig.

Aufgabe Formulieren Sie die im gegebenen Programm enthaltene for-Schleife als

- i) while-Schleife
- ii) do-while-Schleife

Lösungsstruktur Abzugeben ist jeweils ein Programmstück, in dem die for-Schleife durch den verlangten Schleifentyp ersetzt wurde. Das Programmstück soll sich äquivalent auf die Variablen i und a auswirken.

Masstab Die korrekte Implementation mit der while-Schleife wird mit 3 Punkten bewertet, die korrekte Implementation mit der do-while-Schleife wird mit 4 Punkten bewertet. Die gesamte Teilaufgabe ist somit 7 Punkte wert.

Aufgabe 2: Prozeduren und Funktionen: Komplexe Zahlen (25 Punkte)

In dieser Aufgabe geht es um Prozeduren, Funktionen. Wir testen, ob Sie in der Lage sind, Prozeduren und Funktionen zu programmieren, mit Parametern umzugehen und Prozeduren und Funktionen aufzurufen.

Komplexe Zahlen Eine komplexe Zahl $x = a + bi$ wird durch den Realteil $a \in \mathbb{R}$ und den Imaginärteil $b \in \mathbb{R}$ repräsentiert. Dabei ist i die imaginäre Einheit mit der Eigenschaft $i^2 = -1$. In dieser Aufgabe benötigen wir nur folgende Rechenregeln für komplexe Zahlen:

Addition	Multiplikation
$\left. \begin{array}{l} x := a_1 + b_1 i \\ y := a_2 + b_2 i \end{array} \right\} x+y = (a_1+a_2)+(b_1+b_2)i$	$\left. \begin{array}{l} x := a_1 + b_1 i \\ y := a_2 + b_2 i \end{array} \right\} x \cdot y = (a_1 \cdot a_2 - b_1 \cdot b_2) + (a_1 \cdot b_2 + b_1 \cdot a_2)i$
Der Realteil von $x + y$ ist $a_1 + a_2$. Der Imaginärteil von $x + y$ ist $b_1 + b_2$.	Der Realteil von $x \cdot y$ ist $a_1 \cdot a_2 - b_1 \cdot b_2$. Der Imaginärteil von $x \cdot y$ ist $a_1 \cdot b_2 + b_1 \cdot a_2$.

Den Datentyp für eine komplexe Zahl geben wir Ihnen vor:

```
1 struct Complex { // Datentyp fuer eine komplexe Zahl
2     double real; // Realteil
3     double imag; // Imaginaerteil
4 };
```

Aufgaben

- a) Implementieren Sie die Addition von komplexen Zahlen für den vorgegebenen Datentyp `Complex` als Funktion mit Rückgabewert.

Lösungsstruktur Ihre Lösung enthält (1) eine vollständige Funktion, (2) ein Programmfragment eines konkreten *Beispielaufrufs* mit *deklarierten* und *initialisierten* Variablen für die Parameter, (3) das Ergebnis Ihres Beispiels als Zahlenwert und (4) Kommentare.

Masstab Bewertet werden der korrekte Umgang mit Parametern (5 Punkte), die korrekte Deklaration, Initialisation und der Aufruf (3 Punkte), das korrekte Ergebnis (1 Punkt), ein korrekter Anweisungsteil innerhalb der Funktion (1 Punkt) und die Kommentare (1 Punkt). Die Teilaufgabe ist 11 Punkte wert.

- b) Implementieren Sie die Multiplikation von komplexen Zahlen für den vorgegebenen Datentyp `Complex` als Prozedur mit variablen Parametern.

Lösungsstruktur Ihre Lösung enthält (1) eine vollständige Prozedur (2) ein Programmfragment eines konkreten *Beispielaufrufs* mit *deklarierten* und *initialisierten* Variablen für die Parameter, (3) das Ergebnis Ihres Beispiels als Zahlenwert und (4) Kommentare.

Masstab Bewertet werden der korrekter Umgang mit Parametern (4 Punkte), die korrekte Deklaration, Initialisation und der Aufruf (3 Punkte), ein korrekter Anweisungsteil innerhalb der Prozedur (2 Punkt), die Angabe des Ergebnisses(1 Punkt) und die Kommentare (1 Punkt). Die Teilaufgabe ist 11 Punkte wert.

- c) Stellen Sie den Prozeduransatz dem Funktionsansatz gegenüber und erläutern Sie anhand der beiden vorangegangenen Unteraufgaben in drei bis vier Sätzen die Unterschiede zwischen den beiden Ansätzen.

Lösungsstruktur Ihre Lösung enthält drei unterschiedliche Aussagen über die Unterschiede der beiden Ansätze.

Masstab Jede korrekte Aussage wird mit 1 Punkt bewertet. Die Teilaufgabe ist 3 Punkte wert.

Aufgabe 3: Records und Felder (28 Punkte)

Diese Aufgabe prüft Ihre Kenntnisse in Bezug auf Records und Felder. Es wird geprüft ob sie wissen, wie man sie definiert und wie man sie anspricht. Dieses Wissen werden wir anhand des Umrechnens von Einheiten prüfen.

- a) Beim Umrechnen von Einheiten hat man oft mehrere Beziehungen zwischen je zwei Einheiten gegeben und möchte aufgrund dieser Information eine weitere Beziehung zwischen zwei Einheiten herstellen:

Beispiel

Wir wissen:

Beziehung 1

2 yard
=
72 inch

und

Beziehung 2

7.62 cm
=
3 inch

Wir wollen wissen:

Beziehung 3

3 yard
=
??? cm

Dazu berechnen wir

$$3 \text{ yard} = 3 \text{ yard} \cdot \frac{72 \text{ inch}}{2 \text{ yard}} \cdot \frac{7.62 \text{ cm}}{3 \text{ inch}} = 274.32 \text{ cm}$$

Aufgaben

- i) Definieren Sie einen Record *Beziehung*, der eine Box aus dem obigen Beispiel mit Zahlenwerten und Kodierung der Einheiten abbildet.

Lösungsstruktur Ihre Lösung besteht (1) aus einem definierten Recordtyp. (2) Geben Sie den Elementen des Records sinnvolle und einleuchtende Namen. (3) Kommentieren Sie Ihre Lösung.

Massstab Für die korrekte Deklaration erhalten sie 3 Punkte, für die Vergabe von sinnvollen Namen 2 Punkte, Kommentare sind 1 Punkt wert. Die gesamte Teilaufgabe gibt 6 Punkte.

- ii) Schreiben Sie eine Funktion

```
Beziehung umwandlung(Beziehung b1, Beziehung b2, double einheit1)
```

Diese Funktion gibt als Funktionswert eine neue Beziehung aufgrund der Informationen in *b1* und *b2* (siehe auch obiges Beispiel):

- Die Funktion darf davon ausgehen, dass jeweils ein Element in *b1* und *b2* dieselbe Einheit haben.
- Der Parameter *einheit1* ist die neue Masszahl derjenigen Einheit in *b1*, die nicht in *Beziehung b2* vorkommt. Er ist ein Element der neuen Beziehung.
- Das zweite Element der neuen Beziehung enthält das Resultat der Umwandlung gemäss dem obigen Beispiel.

Lösungsstruktur Abzugeben ist eine Lösung bestehend aus (1) einer vollständigen, Funktion, die (2) einen vollständig initialisierten Record gemäss Aufgabenstellung zurückgibt. (3) Kommentieren Sie Ihre Lösung.

Massstab Bewertet werden der Umgang mit Records (7 Punkte), algorithmische Umsetzung (1 Punkt), Beachtung der Funktion (1 Punkt) und Kommentare (1 Punkt). Die gesamte Teilaufgabe gibt 10 Punkte.

⇒

b) Betrachten Sie folgende Umrechnungstabelle für Längeneinheiten:

	cm	m	inch	yard
cm	1	0.01	0.393701	0.0109361
m	100	1	39.3701	1.09361
inch	2.54	0.0254	1	0.027778
yard	91.44	0.9144	36	1

Allgemein ist diese Tabelle so zu lesen

$$\text{Zelleninhalt} = \frac{1 \text{ Zeileneinheit}}{1 \text{ Spalteneinheit}}$$

Beispiel Die erste Spalte der Tabelle liest sich also wie folgt:

$$1 \text{ cm} = 1 \text{ cm}, 1 \text{ m} = 100 \text{ cm}, 1 \text{ inch} = 2.54 \text{ cm}, 1 \text{ yard} = 91.44 \text{ cm}.$$

Problemstellung Wir nehmen nun an, wir haben von einer solchen Tabelle nur die erste Spalte gegeben. Wir möchten die weiteren Werte der Tabelle berechnen.

Vervollständigungsverfahren Die weiteren Einträge der Tabelle erhält man, indem man die Tabelle Spalte für Spalte und innerhalb der Spalte von oben nach unten ergänzt:

- Werte über der Diagonalen erhält man durch Umkehrung des an der Diagonale gespiegelten Wertes.

Beispiel m in inch \Rightarrow inch in m.

$$1 \text{ inch} = 0.0254 \text{ m} \quad \Rightarrow \quad 1 \text{ m} = \frac{1}{0.0254} \text{ inch}$$

- Alle Werte auf der Diagonalen sind 1.
- Werte unter der Diagonalen erhält man durch Multiplikation eines Wertes über der Diagonalen mit dem entsprechenden Wert aus einer früheren Spalte.

Beispiel yard in inch

$$\frac{1 \text{ yard}}{1 \text{ inch}} = \frac{1 \text{ yard}}{1 \text{ cm}} \cdot \frac{1 \text{ cm}}{1 \text{ inch}}$$

Umsetzung in C++ Die Tabelle deklarieren wir in C++ als zweidimensionales Array vom Typ double:

```
double tabelle[N][N]
```

Dabei ist N eine beliebige, nicht näher spezifizierte Konstante, die die Grösse der Tabelle angibt.

Aufgabe Schreiben Sie eine Prozedur

```
void vollstaendig(double tabelle[N][N])
```

Diese Prozedur vervollständigt die durch die erste Spalte gegebene Tabelle `tabelle` mithilfe des angegebenen Verfahrens.

Lösungsstruktur (1) Ihre Lösung enthält eine vollständige Prozedur, welche die übergebene Tabelle vollständig ausfüllt. (2) Versehen Sie Ihre Prozedur mit Kommentaren.

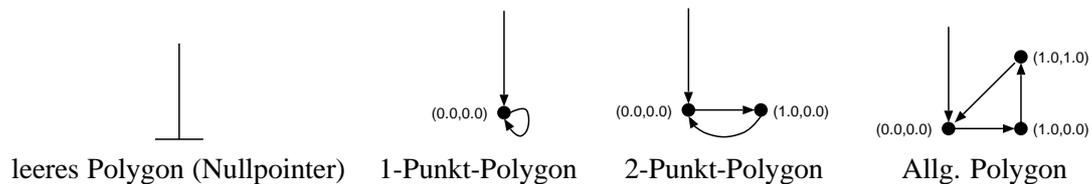
Masstab Bewertet werden die Behandlung von Arrays (8 Punkte), die algorithmische Umsetzung (3 Punkte) sowie Ihre Kommentare (1 Punkt). Die gesamte Teilaufgabe ist 12 Punkte wert.

Aufgabe 4: Dynamische Datenstrukturen (22 Punkte)

In der Vorlesung haben Sie Pointer und dynamische Datenstrukturen kennengelernt. In dieser Aufgabe wird anhand des Beispiels von Polygonen geprüft, ob Sie in der Lage sind, dynamische Datenstrukturen zu definieren und sie zu manipulieren.

Polygone Gegeben sind Polygone in der Ebene. Polygone sind Punktmenge, die durch einen geschlossenen Streckenzug miteinander verbunden sind.

Die folgende Zeichnung zeigt, dass Polygone auf sehr natürliche Weise durch zyklische Listen repräsentiert werden können.



Anhand der Zeichnung sehen Sie auch, dass Punkte in der Ebene ebenfalls als Polygone aufgefasst werden können.

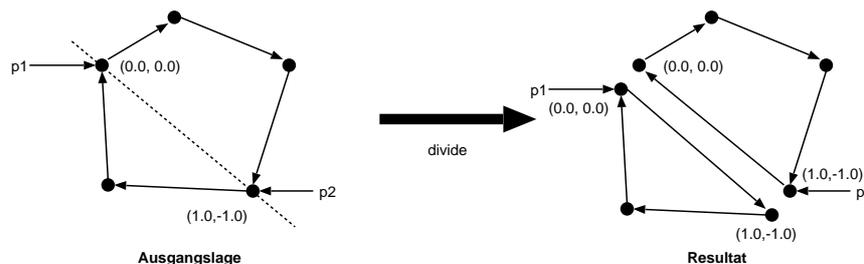
Aufgaben

a) Definieren Sie eine dynamische Datenstruktur `Polygon`.

Lösungsstruktur Ihre Lösung enthält (1) eine vollständige dynamische Datenstruktur zur Darstellung von Polygonen, (2) sinnvolle und einleuchtende Namen, (3) Kommentare zur dynamischen Datenstruktur.

Masstab Ihre Lösung ist gut, wenn sie alle notwendigen Teile für die dynamische Datenstruktur (4 Punkte) und verständliche Kommentare (1 Punkt) enthält. Die Teilaufgabe ist 5 Punkte wert.

b) Gegeben ist *ein* Polygon durch zwei Punkte `p1` und `p2`. Es soll eine Prozedur geschrieben werden, welche das Polygon an der Linie teilt, die durch die beiden Punkte `p1` und `p2` definiert ist. Das eine Resultat soll in `p1` zurückgegeben werden, das zweite Resultat in `p2`. Folgende Zeichnung veranschaulicht die Arbeitsweise der Prozedur:



Aufgabe Schreiben Sie diese Prozedur

```
void Divide(Polygon *p1, Polygon *p2)
```

Sie dürfen dabei davon ausgehen, dass das gegebene Polygon mindestens 2 Punkte (`p1` und `p2`) enthält.

Lösungsstruktur Erwartet werden (1) eine vollständige Prozedur, die die Aufgabenstellung erfüllt und (2) Kommentare zu Ihrer Prozedur. Eine verständliche Zeichnung, aus der hervorgeht, was wann zu tun ist, hat den Stellenwert eines Kommentars.

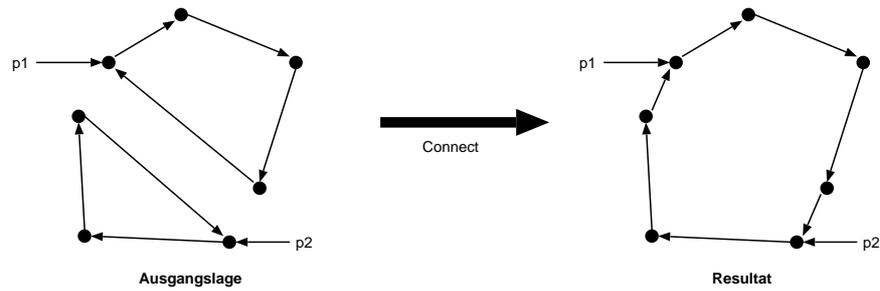
Masstab Bewertet werden der Umgang mit Zeigervariablen (5 Punkte), die Behandlung der dynamischen Datenstruktur (1 Punkt) und Ihre Kommentare (1 Punkt). Die Teilaufgabe ist 7 Punkte wert.

⇒

c) Schreiben Sie in C++ eine Prozedur

```
void connect(Polygon & *p1, Polygon *p2)
```

Diese Prozedur hängt das Polygon p2 an das Polygon p1 an und gibt das resultierende Polygon in p1 zurück. Betrachten Sie dazu auch folgende Zeichnung:



Die beiden gegebenen Polygone sind zyklische Listen, die auch leer sein können.

Lösungsstruktur Erwartet werden (1) eine vollständige, in C++ formulierte Prozedur, die die Aufgabenstellung erfüllt und (2) Kommentare zu Ihrer Prozedur. Eine verständliche Zeichnung, aus der hervorgeht, was wann zu tun ist, hat den Stellenwert eines Kommentars.

Masstab Ihre Lösung ist gut, wenn sie die dynamische Datenstruktur richtig behandelt (7 Punkte) alle möglichen Fälle berücksichtigt (2 Punkte) und Kommentare enthält (1 Punkt). Die Teilaufgabe ist 10 Punkte wert.

Aufgabe 5: Logik (25 Punkte)

In dieser Aufgabe werden Kenntnisse in logischen Variablen und Ausdrücken, Ableitungen von logischen Ausdrücken aus gegebenen Sachverhalten, Wahrheitstabellen und Vereinfachungen von logischen Ausdrücken geprüft.

Sachverhalt Calvin will mit seinem Plüschtiger Hobbes im Garten spielen. Seine Mutter stellt aber Bedingungen:

DU DARFST NICHT IM GARTEN SPIELEN, WENN DAS WETTER SCHLECHT IST UND DU NOCH HAUSAUFGABEN ZU ERLEDIGEN HAST. DU DARFST AUCH NICHT IM GARTEN SPIELEN, WENN DU ZWAR KEINE HAUSAUFGABEN ZU ERLEDIGEN HAST, ABER DAS ZIMMER NOCH AUFGERÄUMT WERDEN MUSS. UND DU DARFST AUCH DANN NICHT IM GARTEN SPIELEN, WENN DAS WETTER ZWAR SCHÖN IST, ABER DU NOCH HAUSAUFGABEN ERLEDIGEN MUSST!



Calvin will nun wissen: Wann darf er überhaupt je im Garten spielen?

Aufgaben

- a) Teilen Sie die Gesamtaussage der Mutter in *elementare* Bedingungen für das Spielen im Garten ein. Deklarieren Sie dazu für jede dieser Bedingungen eine logische Variable in C++ und kennzeichnen Sie mit Kommentaren, wofür diese steht. Es ist Teil der Aufgabe herauszufinden, wieviele elementare Bedingungen die Aussage der Mutter enthält.

Lösungsstruktur Ihre Lösung enthält (1) Deklarationen von logischen Variablen (2) zu jeder Variable ein Kommentar, für welche Bedingung sie steht.

Masstab Bewertet wird, ob Sie in der Lage sind die Aussage der Mutter in elementare Bedingungen zu unterteilen und einander entsprechende Bedingungen zu erkennen. Jede korrekte Deklaration ist 2 Punkte wert.

- b) Schreiben Sie ohne Verwendung der `if`-Anweisung eine Funktion `darfspielen` mit einem Rückgabewert vom Typ `bool`:
- Die Eingabewerte der Funktion sind logische Variablen, die jeweils den Wahrheitswert einer elementaren Bedingung der Mutter speichern.
 - Die Funktion gibt **true** zurück, falls Calvin im Garten spielen darf, ansonsten gibt sie **false** zurück.

Lösungsstruktur Abzugeben ist (1) eine vollständige C++-Funktion, die die gestellte Aufgabe erfüllt und keine `if`-Anweisung enthält, (2) Kommentare zu den Parametern Ihrer Funktion, (3) Kommentare zu den einzelnen Anweisungen Ihrer Funktion.

Masstab Bewertet werden die Zusammensetzung eines logischen Ausdrucks mit 7 Punkten, Kommentare zu Parametern und Kommentare zu den Anweisungen Ihrer Funktion mit jeweils 1 Punkt. Die gesamte Teilaufgabe gibt 9 Punkte.

- c) Erstellen Sie eine Wahrheitstabelle für die in der letzten Unteraufgabe entwickelte Funktion.

Lösungsstruktur Abzugeben ist eine vollständige Wahrheitstabelle.

Masstab Jede korrekte Zeile der Wahrheitstabelle wird mit 1 Punkt bewertet.

- d) Geben Sie (z.B. aufgrund der Wahrheitstabelle) den einfachsten logischen Ausdruck an, der **wahr** ist, wenn Calvin im Garten spielen darf, und **falsch** ist, falls nicht.

Lösungsstruktur Abzugeben ist ein in C++ oder in mathematischer Form notierter logischer Ausdruck.

Masstab Der korrekte Ausdruck wird mit 2 Punkten bewertet.

***** *Viel Erfolg!* *****