

Prüfung — Informatik D-MATH/D-PHYS

26. 01. 2009

09:00–11:00

Dr. Bernd Gärtner, Prof. Juraj Hromkovič

Kandidat/in:

Name:

Vorname:

Stud.-Nr.:

Ich bezeuge mit meiner Unterschrift, dass ich die Prüfung unter regulären Bedingungen ablegen konnte und dass ich die untenstehenden allgemeinen Bemerkungen gelesen und verstanden habe.

Unterschrift:

Allgemeine Bemerkungen und Hinweise:

1. Überprüfen Sie die Vollständigkeit der ausgeteilten Prüfungsunterlagen (zwei beidseitig bedruckte Blätter, bestehend aus 1 Deckseite und 3 Aufgabenseiten mit insgesamt 6 Aufgaben)!
2. Falls Sie während der Prüfung durch irgendeine Behinderung oder Störung beeinträchtigt werden, melden Sie dies bitte sofort der Aufsichtsperson! Spätere Klagen können nicht akzeptiert werden.
3. Erlaubte Hilfsmittel: **Wörterbücher sind erlaubt; sonst keine.**
4. Betrugsversuche führen zu sofortigem Ausschluss und können rechtliche Folgen haben.
5. Pro Aufgabe ist höchstens eine gültige Version eines Lösungsversuches zulässig. Streichen Sie ungültige Lösungsversuche klar durch! Schreiben Sie auf separate Blätter, nicht auf die Aufgabenblätter (ausser es wird explizit verlangt)!
6. Sie dürfen die Aufgaben in beliebiger Reihenfolge lösen. Konzentrieren Sie sich jeweils auf eine Aufgabe, aber teilen Sie sich Ihre Zeit ein!
7. Nach Ablauf der Prüfungszeit verlassen Sie bitte den Raum und lassen Sie nur die Blätter auf Ihrem Platz liegen, die zur Abgabe bestimmt sind! **Diese müssen alle mit Ihrem Namen beschriftet sein. Die Prüfungsblätter sind dabei mit abzugeben!**
8. Die Prüfung ist bestanden, wenn Sie 60 von 120 erreichbaren Punkten erzielen.

1	2	3	4	5	6		Σ

Aufgabe 1. (5 / 10 / 5 Punkte)

- a) Welche der folgenden Aussagen treffen zu (Ja) respektive nicht zu (Nein)? Kreuzen Sie die entsprechenden Antworten direkt auf dem Aufgabenblatt an; Sie müssen keine Begründung angeben.

	Ja	Nein
(i) Moderne, handelsübliche Prozessoren operieren mit einem Befehlsatz, der der Registermaschinen-Sprache, die Sie in der Vorlesung kennen gelernt haben, sehr ähnlich ist. Ein C++ Programm muss, damit es zur Ausführung gelangen kann, zuerst in diese Prozessor-Sprache übersetzt werden.	<input type="checkbox"/>	<input type="checkbox"/>
(ii) Ein Registermaschinen-Programm wird immer Zeile für Zeile ausgeführt. Es gibt keine Möglichkeit, eine Zeile zu überspringen.	<input type="checkbox"/>	<input type="checkbox"/>
(iii) Falls ein Registermaschinen-Programm keinen Befehl <code>If... then...</code> enthält, ist es nicht möglich, dass dieses unendlich lange läuft.	<input type="checkbox"/>	<input type="checkbox"/>
(iv) Ein Algorithmus muss für jede korrekte Eingabe in endlicher Zeit das korrekte Resultat liefern.	<input type="checkbox"/>	<input type="checkbox"/>
(v) Ein Algorithmus ist an eine konkrete Programmiersprache gebunden (z.B. C++) und kann nicht ohne weiteres in eine andere Sprache übertragen werden.	<input type="checkbox"/>	<input type="checkbox"/>

- b) Seien a und b zwei positive ganze Zahlen, die in `Register(1)` und `Register(2)` eingelesen werden. Geben Sie an, was das folgende Programm berechnet. Das heisst, charakterisieren Sie in Abhängigkeit von a und b was in den Zeilen 6 und 7 ausgegeben wird.

1. Read into `Register(1)`
2. Read into `Register(2)`
3. `Register(3) ← 0`
4. `Register(4) ← 1`
5. If `Register(2) ≤ Register(1)`, then go to line 9
6. `Output ← Register(3)`
7. `Output ← Register(1)`
8. End
9. `Register(1) ← Register(1) - Register(2)`
10. `Register(3) ← Register(3) + Register(4)`
11. Go to line 5

- c) Wenn wir zulassen, dass a und b auch 0 sein können, dann kann bei dem Programm aus Teilaufgabe b) ein schwerwiegendes Problem auftreten. Beschreiben Sie das Problem und charakterisieren Sie genau, wann es auftritt.

Aufgabe 2. (6 / 6 / 6 Punkte) Werten Sie die folgenden Ausdrücke von Hand aus und geben Sie dabei alle Zwischenschritte einzeln an (einschliesslich Typenkonversionen und Variablenauswertungen). Befolgen Sie dabei die durch Assoziativität und Präzedenz festgelegte Reihenfolge und wählen Sie die Reihenfolge der übrigen Schritte frei. Die Variable `y` ist vom Typ `bool` und hat den Wert `false`. Bei der Konversion von `bool` zu `int` wird `true` zu 1 und `false` zu 0 umgewandelt.

- a) `3 + 3 / 2 + 3.0 / 2`
- b) `y * 7.0 + (y - 1) * 3.0`
- c) `!! y || (y = true)`

Aufgabe 3. (30 Punkte) Implementieren Sie die folgende Funktion, die testen soll, ob eine Zahl `n` aus aufeinanderfolgenden Primzahlen zusammengesetzt ist. Zum Beispiel ist $105 = 3 \cdot 5 \cdot 7$ eine solche Zahl, weil 3, 5, 7 eine lückenlose Folge von Primzahlen darstellt. Auf der anderen Seite ist zum Beispiel $21 = 3 \cdot 7$ keine solche Zahl, weil die 5 "fehlt". Es spielt dabei keine Rolle, wie oft ein Faktor vorkommt, das heisst $24 = 2^3 \cdot 3$ ist auch eine "aufeinanderfolgende" Zahl. Insbesondere sind natürlich alle Primzahlen und Primzahlpotenzen aufeinanderfolgend. Die 1 sei auch aufeinanderfolgend.

```
// PRE: n > 0
// POST: Der Rueckgabewert ist true g.d.w. die Primfaktorzerlegung
//       von n nur aus aufeinanderfolgenden Primzahlen besteht.
bool aufeinanderfolgend(unsigned int n);
```

Sie können davon ausgehen, dass Sie den folgenden Primtest schon implementiert haben. Das heisst, Sie können die Funktion `prim` in der Funktion `aufeinanderfolgend` verwenden.

```
// PRE: n > 0
// POST: Der Rueckgabewert ist true g.d.w. n eine Primzahl ist.
bool prim(unsigned int n);
```

Aufgabe 4 (8 / 8 Punkte) Im Zusammenhang mit Structs und Klassen haben wir die Schlüsselwörter `public` und `private` kennengelernt.

- a) Beschreiben Sie ganz allgemein den Verwendungszweck dieser Schlüsselwörter. Gehen Sie insbesondere darauf ein, warum es sinnvoll sein kann, das Schlüsselwort `private` zu benutzen.
- b) Illustrieren Sie den Verwendungszweck aus a), indem Sie eine konkrete Struct- oder Klassendefinition angeben, in der mindestens eines der Schlüsselwörter vorkommt, und erklären Sie, *warum* es dort vorkommt.

Aufgabe 5 (10 / 10 Punkte)

- a) Betrachten Sie die folgende Funktion f und geben sie an, was sie berechnet, wenn $e - b$ den Wert 5 hat. Konkret sollen Sie dazu die Werte $b[0], b[1], \dots, b[4]$ nach dem Ablauf der Funktion hinschreiben.

```
// PRE: [b, e) ist ein gueltiger Bereich
void f(unsigned int* b, unsigned int* e)
{
    int n = e - b;
    for (int i = 0; i < n; ++i) {
        b[i] = 1;
        for (int j = i - 1; j > 0; --j)
            b[j] += b[j - 1];
    }
}
```

- b) Geben Sie die Nachbedingung für die Funktion f an. Das heisst, machen Sie in Abhängigkeit von den Argumenten eine allgemeine Aussage, die das Verhalten der Funktion vollständig charakterisiert (die Aussage muss nicht bewiesen werden).

Aufgabe 6 (16 Punkte) Wir haben in der Vorlesung gezeigt, dass es Dezimalzahlen gibt, die keine endliche Binärdarstellung haben, z.B. die Zahl 0.1. Umgekehrt (und auf den ersten Blick vielleicht überraschend) gilt aber: jede Binärzahl hat eine endliche Dezimaldarstellung. Beweisen Sie diese Aussage, oder liefern Sie zumindest ein plausibles Argument. Formal wollen wir uns auf das offene Intervall $(0, 1)$ beschränken und folgendes zeigen: gegeben sei eine k -stellige Binärzahl der Form

$$b = \sum_{i=1}^k b_i 2^{-i}, \quad b_1, b_2, \dots, b_k \in \{0, 1\}.$$

Dann gibt es ein ℓ , so dass b als ℓ -stellige Dezimalzahl geschrieben werden kann; das heisst, b ist von der Form

$$b = \sum_{i=1}^{\ell} d_i 10^{-i}, \quad d_1, d_2, \dots, d_{\ell} \in \{0, 1, \dots, 9\}.$$