Prüfung Informatik D-MATH/D-PHYS 24.1.2011

Dr. Bernd Gärtner, Prof. Juraj Hromkovič

Lösung.

Aufgabe 1.

Die Variable `x` ist vom Typ `int` und hat zu Beginn jeder Auswertung den Wert 2. Die Variable `b` ist vom Typ `bool` und hat den Wert `true`.

Aufgabe	Ausdruck	Typ	Wert
a)	`1.5 / 2 + 3 * 3`	`double`	`9.75`
b)	`x - 1 < 2 || x + 3 == 4 || 3 / x == 1.5`	`bool`	`true`
c)	`!b && 3.7 / 2 * 3 / 5 == 1.11`	`bool`	`false`
d)	`33 * 5 % 7 * 14 / 2 % 7`	`int`	`0`
e)	`2.0f + x++`	`float`	`4.0f`
f)	`3 >= x * 2 || 20 % 7 == 6`	`bool`	`true`
g)	`- x-- - 1 / 2.0`	`double`	`-2.5`

Punktvergabe. Für jeden korrekt genannten Typ gibt es **+1.5 Punkt** und für jeden korrekt genannten Wert gibt es **+1.5 Punkte**. Die Endpunktzahl wird auf den nächsten vollen Zähler aufgerundet. Insgesamt macht das mögliche **+21 Punkte**.

Aufgabe 2.

a) Die Schleife wird *unendlich* oft ausgeführt. Die Werte von `n` vor der `while`-Bedingung sind 16, 9, 5, 3, 2, 2, 2, ...

b) Die Ausgabe sind die Zahlen 3, 9 und 15. Die Zahlen werden hintereinander und ohne Zwischenraum auf den Ausgabestrom geschrieben; also `3915`.

c) Die entsprechenden Deklarationen sollten wie folgt aussehen.

```
int a;
bool b;
double c;
```

d) Der Aufruf `f(3)` terminiert und der Rückgabewert ist 2. Folgendes war zwar nicht gefragt, aber der Vollständigkeit halber sei es erwähnt. Ein Aufruf von `f` terminiert *immer* und der Rückgabewert ist immer 2. Letzteres ist klar, weil es nur eine `return` Anweisung gibt. Ersteres ist mit vollständiger Induktion leicht zu sehen.

e) Nur beim Ausdruck 4 kommt es zu keinem Zeitpunkt zu Kurzschlussauswertung.

f) Die drei Ausdrücke 1, 3 und 4 können mit Hilfe der De Morgan'schen Regel ineinander übergeführt werden. Nicht jedoch der zweite Ausdruck. Ausdruck Nummer 2 ist demnach die richtige Antwort.

g) Da `0.4` im IEEE Standard nicht exakt dargestellt werden kann, ergibt auch die Rechnung `10*0.4` nicht exakt 4. Das Literal `0.125` hingegen kann exakt dargestellt werden. Die Ausgabe ist demnach `AD`.

Punktvergabe. Jede der Teilaufgaben gibt +4 **Punkte**.

a) Das Stichwort Endlosschlaufe gibt +4 **Punkte**

b) Ausgabe enthält die Zahlen 3,9,15 gibt +3 **Punkte**. Das korrekte Ausgabeformat gibt +1 **Punkt**.

c) `int a` gibt +1 **Punkt**, `bool b` gibt +2 **Punkte** und `double c` gibt +1 **Punkt**. `float c` gibt keinen Punkt.

d) Die Aussage, dass der Aufruf terminiert gibt +2 **Punkte**. Der richtige Ausgabewert 2 gibt +2 **Punkte**.

e) Für jeden Ausdruck, der richtig klassifiziert wurde, gibt es +1 **Punkt**.

f) Wenn der Ausdruck Nummer 2 als Aussenseiter erkannt wird, dann gibt es +4 **Punkte**.

g) Für die Aussage, dass zuerst A ausgegeben wird gibt es +2 **Punkte**. Zusätzlich gibt es +2 **Punkte**, falls erkannt wurde, dass als zweites D ausgegeben wird.

Aufgabe 3.

a)
```
// POST: Die Funktion foo bestimmt das Minimum und das Maximum
//          der Elemente im Feld array. Das Minimum wird im Argument
//          d1 und das Maximum wird im Argument d2 zurueck gegeben.
//          Das Feld array und das Argument n werden nicht veraendert.
```

Der Algorithmus funktioniert wie folgt. Es werden immer die nächsten beiden Elemente des Feldes verglichen, die noch nicht angeschaut wurden. Von diesen beiden wird das kleinere mit dem aktuellen Minimum und das grössere mit dem aktuellen Maximum verglichen. Die Zeilen `04` bis `10` dienen dazu, die Invariante zu etablieren, dass in `d1` das Minimum und in `d2` das Maximum aller bis jetzt betrachteten Elemente gespeichert ist.

b) Aus den Zeilen 06 und 15 resultieren $n/2$ Vergleiche. Dann muss jedes Element ausser der ersten beiden noch mit einem der aktuellen Extrema verglichen werden. Dies geschieht in den Zeilen 16, 17, 19 und 20. Das sind noch einmal $n-2$ Vergleiche. Insgesamt sind also $3n/2-2$ Vergleiche nötig.

Punktvergabe.

a) Für eine vollständige Nachbedingung gibt es **+10 Punkte**. Für eine korrekte Beschreibung des Algorithmus können vom Korrektor Teilpunkte vergeben werden.

b) Die Zahl der Vergleiche sollte exakt angegeben werden, da sie auch nicht variabel ist. Für ein gegebenes n ist sie immer gleich. Die richtige Antwort ergibt **+6 Punkte**. Für die richtige Grössenordnung $(3n/2)$ gibt es nur **+4 Punkte**.

Aufgabe 4.
Der Exponent von `b` darf sich von demjenigen von `a` um höchstens 52 unterscheiden, sonst reicht eine Präzision von 53 nicht aus, um `a+b` exakt darzustellen. Dies ist eine notwendige Bedingnung für ein gutes Zahlenpaar.

Für ein fixes `a` gibt es demnach maximal 2 (für das Vorzeichen) $\cdot 2^{52}$ (für die Mantisse) $\cdot 105$ (für den Exponenten) verschiedene `b`, so dass `a+b` gut sein kann. Hingegen gibt es insgesamt $2 \cdot 2^{52} \cdot 2046$ verschiedene Möglichkeiten für `b`. Die guten Paare müssen also in der Minderheit sein.

Punktvergabe. Insgesamt gibt diese Aufgabe **+20 Punklte**. Für die Aussage, dass sich die Exponenten der beiden Zahlen um maximal 52 unterscheiden dürfen, gibt es **+10 Punkte**. Wenn im wesentlichen die oben angeführte Beweisidee vorhanden ist, und man bemerkt, dass 105 deutlich kleiner ist als 2046, gibt es die volle Punktezahl. Natürlich gibt es von dieser Regel kleinere Abweichungen, wenn ungenau oder unvollständig formuliert wurde.

Für die Aussage: “Man verliert, sobald bei der Addition die Kommastelle verschoben wird” gibt es immerhin **+5 Punkte**.

Aufgabe 5.
Die Menge $\mathbb{Q}$ ist abzählbar.

Um dies zu zeigen, müssen wir argementieren, dass es eine Bijektion zwischen $\mathbb{Q}^+$ und $\mathbb{N}$ gibt. Betrachten sie dazu folgende Tabelle.

	1	2	3	4	5	$\ldots$
1	1/1	1/2	1/3	1/4	1/5	
2	2/1	2/2	2/3	2/4	2/5	
3	3/1	3/2	3/3	3/4	3/5	
4	4/1	4/2	4/3	4/4	4/5	
5	5/1	5/2	5/3	5/4	5/5	
$\vdots$						$\ddots$

Um eine Bijektion zu erstellen, kann die Tabelle nun entlang der positiven Diagonalen traversiert werden, d.h. wir bringen die rationalen Zahlen in die Reihenfolge: 1/1, 2/1, 1/2, 3/1, 2/2, 1/3, 4/1, 3/2, 2/3, ... Sie sehen jetzt schon, dass in dieser Liste alle rationalen Zahlen mehrfach vorkommen. Mehrfachnennungen werden bei der Nummerierung einfach ausgelassen. Wir ordnen jeder noch nicht gepaarten rationalen Zahl die entsprechende natürliche Zahl in dieser Reihenfolge zu. Die Abbildung ist injektiv, weil wir niemals die selbe natürliche Zahl verwenden. Die Abbildung ist surjektiv, weil wir letztendlich jede natürliche Zahl verwenden.

Um nun auf die Abzählbarkeit von ganz $\mathbb{Q}$ zu schliessen, kann man bemerken, dass die oben angeführte Konstruktion genau so für $\mathbb{Q}^-$ funktioniert. Man kann die beiden Tabellen zum Beispiel alternierend traversieren. Die Reihenfolge, die sich dann ergibt, wäre $1/1$, $-1/1$, $2/1$, $-2/1$, $1/2$, $-1/2$, $3/1$, $-3/1$, ...

Diese Bijektion explizit anzugeben, ist nicht mehr so einfach und war nicht gefragt.

Punktvergabe. Wenn nur die Abzählbarkeit von $\mathbb{Q}^+$ gezeigt wird, gibt es bereits 12 Punkte, für die Abzählbarkeit von ganz $\mathbb{Q}$ die restlichen 3.

+12 Punkte wenn die Abzählbarkeit von $\mathbb{Q}^+$ gezeigt wird.

Zum Beispiel durch die Beschreibung einer bijektiven Abbildung zwischen $\mathbb{N}$ und $\mathbb{Q}^+$ (z.B. "Diagonalverfahren"). Es muss gezeigt werden, dass die angegebene Abbildung bijektiv ist, -2 **Punkte** pro fehlende Richtung.

Alternativ kann man auch nur eine surjektive Abbildung von $\mathbb{N}$ auf $\mathbb{Q}^+$ angegeben werden. Es muss gezeigt werden, dass die Abbildung surjektiv ist und weshalb dies genügt, -2 **Punkte** pro fehlende Richtung.

Das Aufzeichnen einer Tabelle, die alle Zahlen aus $\mathbb{Q}^+$ beinhaltet, genügt nicht. Die Abbildung/Abzählung muss durch Worte oder durch (mindestens) Pfeile auf der Skizze explizip angegeben werden.

+3 Punkte wenn die Abzählbarkeit von ganz $\mathbb{Q}$ gezeigt wird.

Entweder duch einen direkten Beweis, oder duch die Angabe eines Grundes, weshalb man von der Abzählbarkeit von $\mathbb{Q}^+$ auf die Abzählbarkeit von $\mathbb{Q}$ schliessen kann. -2 **Punkte** wenn keine Abbildungsvorschrift beschrieben wurde (z.B. "alternierend positive und negative Zahlen" genügt, aber "es gibt gleich viele positive wie negative Zahlen" zählt nicht).

Formale Abbildungsvorschriften wurden nicht verlangt.

Argumente, bei denen auf die Vorlesung verwiesen wird, z.B. “wir wissen, dass $\mathbb{Q}^+$ abzählbar ist”, geben keine Punkte.

Argumente, die auf Lemmas (ohne Beweis) verweisen, genügen nicht als Lösung und geben höchstens Teilpunkte. Insbesondere die Verwendung von “Vereinigung abzählbar vieler abzählbarer Mengen ist abzählbar” oder “$\mathbb{N} \times \mathbb{N}$ ist abzählbar” zählen nicht, da dies im wesentlichen in dieser Aufgabe hätte gezeigt werden müssen.

Aufgabe 6.
Die Funktion lässt einfach zwei Zeiger von links und von rechts aufeinander zuwandern. So lange sie sich noch nicht passiert haben, werden die entsprechenden beiden Elemente jeweils vertauscht. Um die ursprünglichen Zeiger `begin` und `end` nicht zu verändern, sollten Kopien gemacht werden und Hilfszeiger verwendet werden.

Beachten sie, dass es in der folgenden Implementierung nicht darauf an kommt, ob das Feld eine gerade oder ungerade Länge hat.

```
void reverse(int* begin, int* end) {
  if (end - begin > 0) {
    int* l = begin;
    int* r = end;
    while (r - l > 0) {
      int temp = *l;
      *l = *r;
      *r = temp;
      ++l;
      --r;
    }
  }
}
```

Punktvergabe. Die Punktevergabe ist noch vom Korrektor zu bestimmen. Bitte noch Angaben machen. Insgesamt gibt es **+20 Punkte**. Es sollte darauf geachtet werden, dass die Implementierung für gerade und ungerade Längen funktioniert. Falls das nicht der Fall sein sollte gibt es **-3 Punkte**. Ebenfalls sollte `begin` und `end` nicht verändert werden. Auch das gibt **-3 Punkte** falls nicht erfüllt.

Grundsätzlich gilt ein Syntaxfehler wird verziehen, aber ab dem zweiten gibt es je **-1 Punkt** Abzug. Die Wiederholung des selben Fehlers gilt nicht als erneuter Fehler.

Ende der Musterlösung zur Prüfung vom 24.1.2011.