2. Übungsblatt
Thema: Entwurf von Algorithmen

Aufgabe 8

Schreiben Sie ein Programm, das den größten gemeinsamen Teiler (*ggT*) zweier natürlicher Zahlen nach dem *Euklidischen Algorithmus* (Euklid, Die Elemente, 7. Buch, §2) ermittelt.
Das Euklidische Verfahren beruht auf der Tatsache, daß, wenn *u* größer als *v* ist, der *ggT* von *u* und *v* gleich dem *ggT* von *v* und *u*-*v* ist (Beweis: siehe Euklid, Die Elemente).
Beispiel: ggT(461 952, 116 298) = ggT(116 298, 345 654) = ggT(116 298, 229 356) =
ggT(113 058, 116 298) = ggT(113 058, 3240) = ...
Überlegung: Wenn $u > v$ gilt, wird so lange *v* von *u* subtrahiert, bis eine Zahl herauskommt, die kleiner als *v* ist. Diese Zahl ist gleich dem Rest, der bei der Division von *u* durch *v* bleibt. Verbessern Sie den obigen Algorithmus durch Verwendung des Restoperators.
Beispiel: ggT(461 952, 116 298) = ggT(116 298, 113 058) = ggT(113 058, 3 240) =
ggT(3 240, 2 898) = ggT(2 898, 342) = ggT(342, 162) = ggT (162, 18) = ...

Aufgabe 9

Schreiben Sie ein Programm, das prüft, ob eine gegebene Zahl eine Primzahl ist. Berücksichtigen Sie dabei die Eigenschaft, daß eine Primzahl nur durch eins und durch sich selber teilbar ist. Als erster Lösungsansatz ist zu untersuchen, ob die Zahl durch einen ihrer Vorgänger teilbar ist. Überlegen Sie sich zu dieser Vorgehensweise Verbesserungen, die die Effizienz des Programms steigern!

Aufgabe 10

Eine Zahl $n \in \mathbb{N}$ heißt *perfekt* oder *vollkommen*, wenn die Summe ihrer Teiler gleich *n* ist (Euklid, Die Elemente, 7. Buch, Def. 22). Dabei sind die Teiler echt kleiner als *n*. Beispielsweise ist 6 = 1 + 2 + 3 eine perfekte Zahl.
a) Geben Sie einen imperativen Algorithmus an, der für eine gegebene Zahl bestimmt, ob sie perfekt ist.
b) Erweitern Sie das Programm derart, daß alle perfekten Zahlen bis 1000 bestimmt werden.

Aufgabe 11

Entwerfen Sie einen Algorithmus zum Umwandeln von Zahlen in verschiedene Zahlensysteme.
a) Lesen Sie eine ganze Zahl ein. Wandeln Sie diese in eine Dualzahl, eine Oktalzahl und eine Hexadezimalzahl um.
b) Lesen Sie eine Dualzahl ein und wandeln Sie diese in eine Oktalzahl, eine Dezimalzahl und eine Hexadezimalzahl um.
Benutzen Sie zur Lösung dieser Aufgaben keine Standardfunktionen.

Aufgabe 12

Schreiben Sie ein Programm, das die Anzahl der Möglichkeiten zum Wechseln eines beliebigen Geldbetrages berechnet.
Gehen Sie dabei von den in Deutschland üblichen Münzen und Geldscheinen aus. Beschreiben Sie zunächst Ihren Lösungsansatz in natürlicher Sprache.

Aufgabe 13

Beschreiben Sie in natürlicher Sprache einen Algorithmus zum Suchen einer natürlichen Zahl in einer
a) unsortierten Folge und in einer
b) sortierten Folge.

Thema: Applikative Algorithmen

Aufgabe 14

Welche Funktion wird durch die folgende Definition realisiert? Beschreiben Sie die Auswertung an einem Beispiel.

f(x, y) = if y = 0 then x
else f(x - 1, y - 1) fi

Aufgabe 15

Für die Funktion *f* gelte $f(x, y) * ggT(x, y) = x * y$ mit $x, y \in \mathbb{N}, x, y > 0$.

a) Geben Sie, ohne dabei *ggT* zu benutzen, einen applikativen Algorithmus zur Berechnung von *f* an.
b) Unter welcher Bezeichnung ist *f* allgemein bekannt?

Thema: Registermaschine

Aufgabe 16

Gegeben sei eine Registermaschine mit dem folgenden Programm

```
 1  CLOAD 0
 2  STORE 2
 3  LOAD 2
 4  MULT 2
 5  STORE 3
 6  LOAD 1
 7  SUB 3
 8  If c0 = 0 GOTO 12
 9  LOAD 2
10  CADD 1
11  GOTO 2
12  END
```

a) Geben Sie die Folge der Konfigurationen an, die bei der Abarbeitung des Programms beginnend mit (1, 0, 15, 0, 0, ...) entsteht (Beschränkung auf die ersten vier Speicherregister.).
b) Bestimmen Sie die von der Registermaschine durch Ausgabe im ersten bzw. im zweiten bzw. dritten bzw. vierten Speicherregister berechneten einstelligen Funktionen.

Aufgabe 17

Bestimmen Sie die von den Registermaschinen mit den folgenden Programmen berechneten zweistelligen Funktionen, wobei das Ergebnis stets im ersten Speicherregister enthalten sein soll.

```
 1  LOAD 2
 2  if c0 = 0 GOTO 10
 3  LOAD 1
 4  CADD 1
 5  STORE 1
 6  LOAD 2
 7  CSUB 1
 8  STORE 2
 9  GOTO 2
10  END
```

```
 1  LOAD 2
 2  CSUB 5
 3  if c0 = 0 GOTO 8
 4  LOAD 2
 5  MULT 1
 6  STORE 1
 7  GOTO 11
 8  LOAD 2
 9  ADD 1
10  STORE 1
11  END
```

Beschreiben Sie die berechneten Funktionen durch Konstrukte einer Programmiersprache.