

Übungsblatt 1 zu Grundlagen der Informatik I

Michael Kopp

Bis 6. November 2008 abgeben

Aufgabe 1.1

Für die Zahlen a und b berechnet der folgende Algorithmus den ggT:

```
while(b != 0){
    h = a % b;
    a = b;
    b = h;
}
print "der ggT ist a \n";
```

Es ergibt sich:

Zahl 1	Zahl 2	ggT
23	42	1
15736	10876	4
65536	524288	65536
16	47123	1

Aufgabe 1.2 – Zahlenkonversion

Allgemeine Umrechnungsvorschriften

Binär – Dezimal Horner-Schema: von links nach rechts wird in den Taschenrechner eingegeben (angefangen wird bei der ersten 1 – hier gibt man eine 1 ein und drückt [=]): für eine 1 ein $\times 2 + 1$ und für eine 0 nur eine $\times 2$.

Dezimal – Binär Man teilt die Dezimalzahl durch 2. Ist der Teilerrest (modulo) 1, schreibt man eine 1, ist er 0, schreibt man eine 0. Die Zahl, die man als letztes ausrechnet, steht am Anfang der Binärzahl und man teilt so lange, bis man als Ergebnis 0 stehen hat.

Binär – Oktal Man teilt die Binärzahl in Dreiergruppen ein (man beginnt rechts). Für jede 3er-Gruppe ermittelt man den Wert in Oktal (eine Ziffer zwischen 1 und 7) und schreibt diesen Wert für die Dreiergruppe auf.

Oktal – Binär Entsprechend der Rückweg: Für jede Oktalziffer wird eine 3er-Gruppen von Binärzahlen aufgeschrieben.

Binär – Hexadezimal Wie bei Binär–Oktal, nur dass man 4er-Gruppen abteilt

Hexadezimal – Binär Ebenfalls der Rückweg

Dezimal – Hexadezimal Man teilt die Zahl durch 16. Den Teilerrest schreibt man auf, das Divisionsergebnis verwendet man weiter – es wird wieder durch 16 geteilt usw.

Dezimal – Oktal Wie Dezimal – Hexadezimal nur mit 8 statt 16

Hexadezimal – Oktal bzw. Oktal – Dezimal Umweg über die Binärdarstellung

Ein Tip zur Schreibweise

1. Die Zahl Schreiben und durch die Basis der Zahl teilen, in deren System umgerechnet werden soll
2. Das (ganzzahlige) Teilergebnis aufschreiben
3. Den Teilerrest *darunter* schreiben
4. Die fertige Zahl liest sich dann von rechts nach links in der Zeile der Teilerreste.

Bsp.:

135 in Binär:

135	67	33	16	8	4	2	1	0
	1	1	1	0	0	0	0	1

Ergebnis: 10000111

Beispiele

- $(1010\ 0101\ 1101\ 1111)_2 = (42463)_{10}$
- $(1111\ 0100\ 1001\ 0000)_2 = (62608)_{10}$
- $(16804)_{10} = (10\ 1001\ 0101\ 1111)_2$
- $(35601)_{10} = (1000\ 1011\ 0001\ 0001)_2$
- $(34981)_{10} = (1000\ 1000\ 1010\ 0101)_2$
- $(43981)_{10} = (1010\ 1011\ 1100\ 1101)_2 = (ABCD)_{16}$
- $(4681)_{10} = (1\ 001\ 001\ 001\ 001)_2 = (11111)_8$

Aufgabe 1.3 – Algorithmus

Fahrkartenautomat

- Eingabe01: Auswahl von Zielort
- Eingabe02: Auswahl von Kartenart (Einzel- Gruppenkarte)
- Eingabe03: Auswahl von Zusätzen (Fahrrad-, Hunde-, etc -karte)
- Ausgabe01: Eingaben zusammenfassen, Preis
- Sensoren01: Wird Hartgeld-, Karte-, Scheineinwurf verwendet
- Geld annehmen
 - Wenn Schein / Karte / Münze ungültig: goto: Ausgabe01
- Rechnung01: Geld, angenommen - Geld zu zahlen = Geld auszugeben
- Rechnung02: Zusammenstellung des Geldes (Münzenarten etc)
- Geld ausgeben, Karte drucken

Kochrezept

Für einen Algorithmus gilt allgemein:

1. Er ist in endlich vielen Schritten lösbar
2. er besteht aus mehreren (Rechen)Schritten
3. einzelne Abschnitte können mehrfach durchlaufen werden

Dies ist auch für ein Kochrezept zu sagen.

```
#define Zutat1 2_Eier
#define Zutat2 200g_Butter
#define Zutat3 150g_Zucker

(Eigelb,Eiweiß) = Zutat1.trennen();
Schüssel1.Inhalt = Eiweiß;
Schüssel1.rühren();
wait(1min);
Schüssel1.zugeben(Zutat3/2);
Schüssel2.Inhalt = Eigelb;
Schüssel2.rühren();
Schüssel2.zugeben(Zutat3/2);
Schüssel2.zugeben(Zutat2);
```

```
wait(2min);  
Schüssel2.zugeben(Schüssel1.Inhalt);  
Schüssel2.rühren;  
wait(1min);  
return Schüssel2.Inhalt;
```