Aufgabe 1

Umwandlung von $(420,42)_{10}$ in eine Festkommazahl zur Basis 2 mit 10 Vorkomma- und 6 Nachkommastellen:
Zuerst die Umwandlung von $(420)_{10}$ in eine Festkommazahl:

$$\begin{array}{ll}
420 : 2 = 210 & \text{Rest } 0 \\
210 : 2 = 105 & \text{Rest } 0 \\
105 : 2 = 52 & \text{Rest } 1 \\
52 : 2 = 26 & \text{Rest } 0 \\
26 : 2 = 13 & \text{Rest } 0 \\
13 : 2 = 6 & \text{Rest } 1 \\
6 : 2 = 3 & \text{Rest } 0 \\
3 : 2 = 1 & \text{Rest } 1 \\
1 : 2 = 0 & \text{Rest } 1
\end{array}$$

Nun folgt die Umwandlung von $(0,42)_{10}$ in eine Festkommazahl:

$$\begin{array}{ll}
0,42 \cdot 2 = 0,84 & \text{1. Ziffer ist } 0 \\
0,84 \cdot 2 = 1,68 & \text{2. Ziffer ist } 1 \\
0,68 \cdot 2 = 1,36 & \text{3. Ziffer ist } 1 \\
0,36 \cdot 2 = 0,72 & \text{4. Ziffer ist } 0 \\
0,72 \cdot 2 = 1,44 & \text{5. Ziffer ist } 1 \\
0,44 \cdot 2 = 0,88 & \text{6. Ziffer ist } 0 \\
0,88 \cdot 2 = 1,76 & \text{7. Ziffer ist } 1
\end{array}$$

Also ergibt sich insgesamt: $(420,42)_{10} = (110100100,0110101)_2$.

Mit Aufrunden ergibt sich für die Festkommadarstellung mit 10 Vor- und 6 Nachkommastellen zur Basis 2:

$$(420,42)_{10} = (0110100100,011011)_2$$

.

Aufgabe 2

Zuerst umwandeln von 181 und 93 in Dualzahlen:
Wir benutzen das Verfahren der wiederholten Division mit Rest:

$$\begin{array}{ll}
181 : 2 = 90 & \text{Rest } 1 \\
90 : 2 = 45 & \text{Rest } 0 \\
45 : 2 = 22 & \text{Rest } 1 \\
22 : 2 = 11 & \text{Rest } 0 \\
11 : 2 = 5 & \text{Rest } 1 \\
5 : 2 = 2 & \text{Rest } 1 \\
2 : 2 = 1 & \text{Rest } 0 \\
1 : 2 = 0 & \text{Rest } 1
\end{array}$$

Also ergibt sich: $(181)_{10} = (10110101)_2$.

$$\begin{array}{ll}
93 : 2 = 46 & \text{Rest } 1 \\
46 : 2 = 23 & \text{Rest } 0 \\
23 : 2 = 11 & \text{Rest } 1 \\
11 : 2 = 5 & \text{Rest } 1 \\
5 : 2 = 2 & \text{Rest } 1 \\
2 : 2 = 1 & \text{Rest } 0 \\
1 : 2 = 0 & \text{Rest } 1
\end{array}$$

Also ergibt sich: $(93)_{10} = (1011101)_2$.

Zahlen anpassen und Komplementbildung:
Die zweite Zahl ist um eine Stelle kürzer als die erste Zahl. Deshalb ist zu Beginn der zweiten Zahl eine 0 zu ergänzen. Da wir im 2-Komplement rechnen wollen und wir die erste Zahl sonst auf Grund ihrer führenden 1 als negative Zahl interpretieren würden, müssen wir bei beiden Zahlen eine weitere führende 0 ergänzen.
Wir erhalten also: $(181)_{10} = (010110101)_{2K}$ und $(93)_{10} = (001011101)_{2K}$
Nun bestimmen wir $(-93)_{10}$ im 2-Komplement durch Komplementbildung (Kippen der Bits und Addition einer 1).

$$\begin{array}{r}
0\,0\,1\,0\,1\,1\,1\,0\,1 \\
\hline
1\,1\,0\,1\,0\,0\,0\,1\,0 \\
+\ 0\,0\,0\,0\,0\,0\,0\,0\,1 \\
\hline\hline
1\,1\,0\,1\,0\,0\,0\,1\,1
\end{array}$$

Es ergibt sich $(-93)_{10} = (110100011)_{2K}$.

Addition im 2-Komplement:
Nun addieren wir $(181)_{10} = (010110101)_{2K}$ und $(-93)_{10} = (110100011)_{2K}$ um die Subtraktion durchzuführen.

$$\begin{array}{rr} & 0\,1\,0\,1\,1\,0\,1\,0\,1 \\ + & 1_1 1\,0_1 1\,0\,0_1 0_1 1_1 1 \\ \hline\hline & 1\,0\,0\,1\,0\,1\,1\,0\,0\,0 \end{array}$$

Da wir im 2-Komplement rechnen betrachten wir auch nur die 9 niederwertigsten Bit des Ergebnisses, d.h. wir erhalten $(010110101)_{2K} + (110100011)_{2K} = (001011000)_{2K}$.

Dezimaldarstellung des Ergebnisses:
Das Ergebnis $(001011000)_{2K}$ ist eine positive Zahl und wir können wie in Aufgabe 1 (c) die Dezimaldarstellung bestimmen. Es gilt:

$$\begin{aligned} (001011000)_2 &= 0 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \\ &= 0 + 0 + 64 + 0 + 16 + 8 + 0 + 0 + 0 \\ &= 88 \end{aligned}$$

Alternativ mit dem Hornerschema.

Aufgabe 3

Konsoleneingabe und -ausgabe:

```
$gcc -o main *.c
$main

a = 200, b = -56, hex = ffffffc8, c = 56
```

Interpretation pro Zeile:

```c
// reserviere 2 unsigned 8 Bit Zeichen mit Wertebereich [0,...,255]
unsigned char a,c;

// reserviere 1 signed 8 bit Zeichen, Wertebereich [-128,...,+127]
signed char b;

// belege a mit 200 dezimal = (11001000) binaer
a=200;

// Kopiere den Inhalt von a nach b => b = (11001000) binaer,
// das ist aber -56 in signed 8 bit
b=a;

// c= B^8 - b = 2-Komplement-Darstellung von
// b = -56 => -(-56) = 56 => c = 00111000
c=256-b;

// Ausgabe dezimal 200, -56, hex = ffffffc8 wg 32 Bit,
// (aber 0Xc8 wuerde reichen), 56
printf("a = %d, b = %d, hex = %x, c = %d\n", a, b, b, c);
```