Aufgabe 1

Für den Ausdruck

$$(x_0 + x_1) \cdot (x_0 + \overline{x_1}) \quad x_0, x_1 \in A$$

mit $A = \{0, 1\}$ ergibt sich folgende Wahrheitstabelle:

x_1	x_0	$y = x_0 + x_1$	$z = x_0 + \overline{x_1}$	$y \cdot z$
0	0	0	1	0
0	1	1	1	1
1	0	1	0	0
1	1	1	1	1

Durch die Gleichheit des Ausdrucks mit der Eingabevariable x_0 für alle möglichen Eingabekombinationen ergibt sich

$$(x_0 + x_1) \cdot (x_0 + \overline{x_1}) = x_0$$

Für die Bearbeitung der Hausaufgabe beachten Sie bitte: Da die Grundmenge einer Booleschen Algebra im Allgemeinen nicht nur aus zwei Elementen bestehen muss, reicht für den allgemeinen Fall eine Wahrheitstabelle nicht aus. Für die Bearbeitung der Hausaufgaben sollten Sie sich deshalb an folgender Lösung orientieren, die auch für eine beliebige Boolesche Algebra gilt:

$$\begin{aligned}(x_0 + x_1) \cdot (x_0 + \overline{x_1}) &\overset{2malA1}{=} (x_1 + x_0) \cdot (\overline{x_1} + x_0) \overset{A3}{=} (x_1 \cdot \overline{x_1}) + x_0 \\ &\overset{A1}{=} \underbrace{(\overline{x_1} \cdot x_1)}_{\overset{A4}{=}0} + x_0 \overset{A2}{=} x_0.\end{aligned}$$

(Die A1 - A4 über den Gleichheitszeichen stehen für die verwendeten Axiome aus der Boole'schen Algebra).

Aufgabe 2

(a) Für die Überprüfung der eingegebenen Zahl `eingabe` ergibt sich folgender Programmablaufplan:

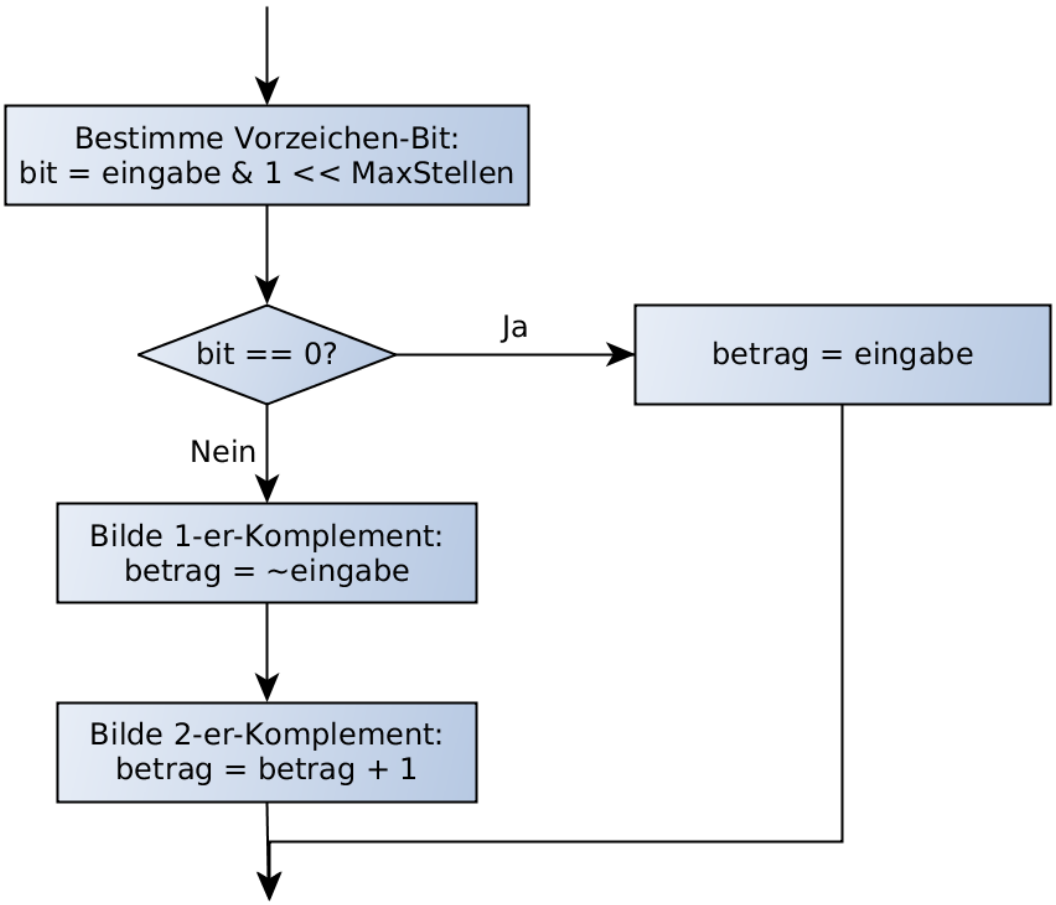

Abbildung 1: Programmablaufplan für (a)

(b) Programmablaufplan für die Berechnung der signifikanten Stellen der zu untersuchenden Zahl

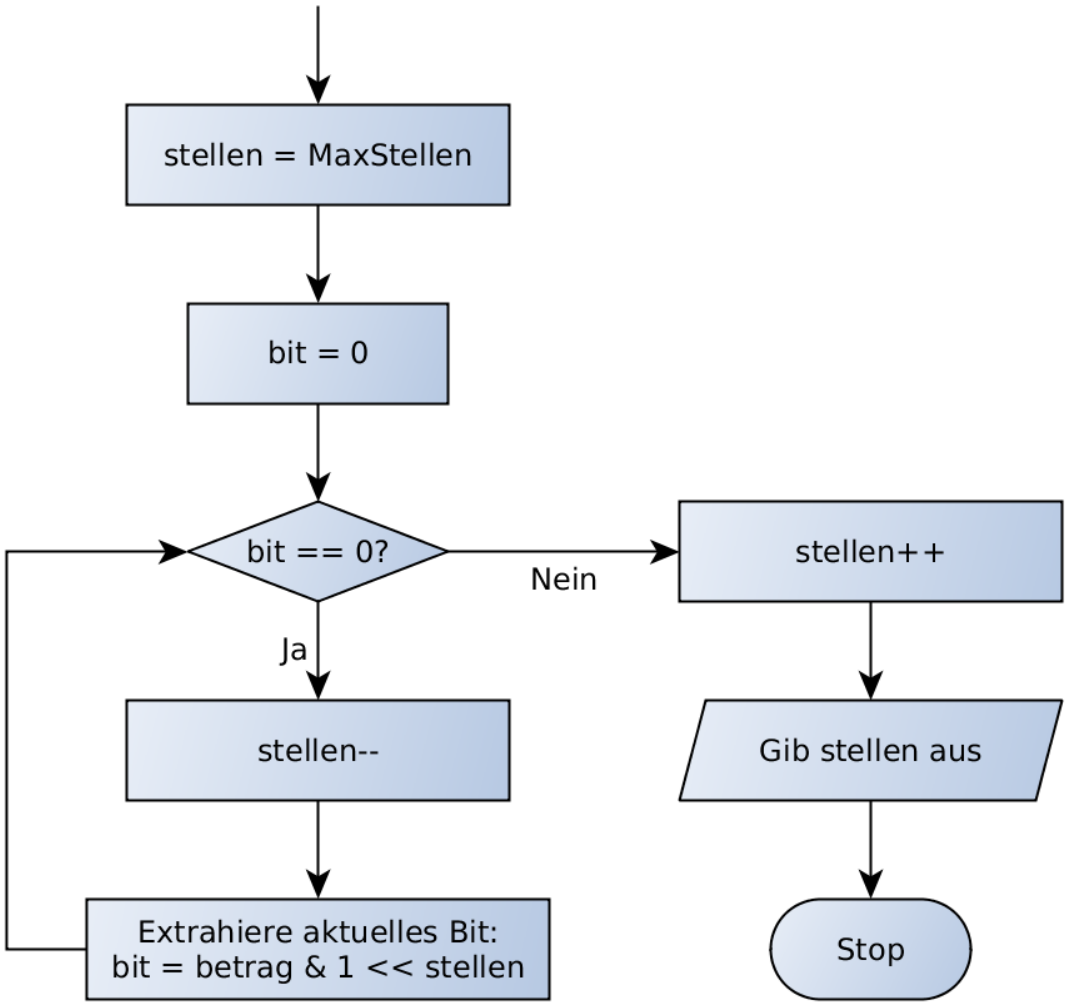

Abbildung 2: Programmablaufplan für (b)

```c
#include <stdio.h>

int main()
{
  /* Gib die Anzahl der signifikanten Stellen des Betrags einer ganzen Zahl aus
     eingabe: zu analysierende Zahl, betrag: positiver Wert von eingabe,
     stellen: Anzahl der signifikanten Stellen von betrag als Ausgabe,
     maxStellen: maximal moegliche Stellenanzahl von betrag als Integerzahl
   */

  int eingabe, betrag, stellen, maxStellen;
  printf("Gib ganze Zahl ein: ");
  scanf("%d",&eingabe);

  maxStellen= sizeof(int) << 3; // Berechne Bitanzahl durch Byteanzahl * 8

  if (eingabe==0)
  {
    printf("Sonderfall Eingabe=0, keine Auswertung\n");
    return 0;
  }

  // Jetzt Studentencode Aufgabe a)
  int bit;

  // Test auf Vorzeichen in MSB

  maxStellen--; // Beginne Zaehlung maxStellen-1 bis 0

  bit = eingabe & 1 << maxStellen;  // Selektiere bit MSB

  if (bit==0)
  {
    betrag=eingabe; // Eingabe war schon positiv, betrag=eingabe
    printf("Zahl ist positiv!\n");
    printf("MSB: 0X%X\n",bit);
  }
  else
  {
    bit=1; // wenn eine 1 in MSB, dann ist Eingabe negativ
    printf("Zahl ist negativ!\n");
    printf("MSB: 0X%X\n",bit);

    // Konvertiere negative Zahl in Betrag mit 2-Komplement
    printf("Zahl = %d,  %x\n", eingabe ,eingabe);
    betrag = ~eingabe;  // 1-Komplement
    printf("1-Komplement= %d, 0X%X\n", betrag, betrag);

    betrag++;
    printf("Betrag (2-Komplement)= %d, 0X%X\n", betrag, betrag);
```

```
50    }
51
52    // Code Aufgabe a) zu Ende
53
54    // Studentencode Aufgabe b)
55    // Berechne signifikante Stellen ueber das Auffinden des hoechstwertigen
          Bits mit Wert 1
56    // Hierzu maskiere einzelne Bits absteigend, bis eines ungleich 0 ist
57
58    stellen = maxStellen; // starte bei MSB-Stelle, die ist sicher 0
59    bit = 0;
60
61    while (bit==0)
62    {
63      stellen--; // naechste Stelle testen
64      bit = betrag & 1 << stellen; // wenn bit !=0, echtes MSB gefunden, sonst
          weiter
65    }
66
67    stellen++; // Anahl der sig. Stellen festlegen
68
69    // Code Aufgabe b) zu Ende
70
71    printf("Zahl hat %d signifikante Stellen\n",stellen);
72    return 0;
73  }
```