


```

59 set1234 :: Int -> Int -> Int -> Int
60 set1234 a b c
61     | a + b + c == 6 = 4           -- Sind a,b und c gleich 1, 2 und
62                                     -- 3 und addiert man diese
63                                     -- Integer erhält man 6 und da die
64                                     -- Addition von 1, 2 und 3 die
65                                     -- einzige Möglichkeit ist aus 3
66                                     -- Werten aus der Menge die Zahl 6
67                                     -- zu erhalten, kann man daraus
68                                     -- schließen, dass der fehlende
69                                     -- Wert, der ausgegeben werden
70                                     -- soll 4 ist.
71     | a + b + c == 7 = 3           -- Sind a, b und c gleich 1,
72                                     -- 2 und 4 erhält man bei der
73                                     -- Addition der 3 Zahlen 7 und
74                                     -- kann daraus schließen, dass der
75                                     -- fehlende Wert 3 ist.
76     | a + b + c == 8 = 2           -- Sind a, b und c gleich 1,
77                                     -- 3 und 4 erhält man bei der
78                                     -- Addition der 3 Integer 8 und
79                                     -- kann daraus schließen, dass
80                                     -- der fehlende Wert 2 ist.
81     | a + b + c == 9 = 1           -- Sind a, b und c gleich 2, 3
82                                     -- und 4 erhält man bei der
83                                     -- Addition der 3 Integer 9 und
84                                     -- kann daraus schließen, dass
85                                     -- der fehlende Wert 1 ist.
86     | otherwise = 0                -- Trifft keiner der 4 Fälle zu,
87                                     -- hat die Funktion den Ausgabe-
88                                     -- parameter 0.
89
90 valueAt :: Float -> Float -> Float -> Float
91 valueAt x y z = (y / (-x)) * z + y -- Die allgemeine Gleichung einer Gerade ist
92                                     --  $y = m \cdot x + n$ . Die Steigung m errechnen wir
93                                     -- indem wir Die Differenz von y2 und y1
94                                     -- ausrechnen und diese Differenz durch die
95                                     -- Differenz von x2 und x1 teilen. Y2 ist
96                                     -- ein Eingabeparameter, Y1 ist 0, da ein
97                                     -- Schnitt mit der X-Achse vorliegt. X2 ist
98                                     -- 0, da die X-Achse geschnitten wird und X1
99                                     -- ist ein Eingabeparameter. n ist in
100                                    -- diesem Fall Y2, da n den Schnitt mit der
101                                    -- Y-Achse darstellt. Gibt man nun einen
102                                    -- weiteren Wert x ein, erhält man den Wert
103                                    -- y, so dass der Punkt (x,y) auf g liegt.
104
105 testParallel :: Float -> Float -> Float -> Float -> Bool
106 testParallel x y a b
107     | (y / (-x)) == (b / (-a)) = True -- Damit eine 2 Geraden Parallel zueinander
108                                         -- sind, müssen die Steigungen gleich sein.
109                                         -- Die Berechnung der Steigung ist bereits
110                                         -- in 2a) beschrieben worden. Sind die
111                                         -- Steigung der Geraden g und h gleich, dann
112                                         -- ist der Ausgabeparameter True.
113     | otherwise = False                -- Sind die Steigungen unterschiedlich, ist
114                                         -- der Ausgabeparameter False.
115
116 parallelThroughX :: Float -> Float -> Float -> Float

```

```

117 parallelThroughX x y a = (-a) * (y / (-x)) -- Damit die Geraden parallel zu-
118 -- einander sind, müssen beide
119 -- Steigungen wieder gleich sein.
120 -- Aus den Eingabeparametern
121 -- errechnet man die Steigung für
122 -- die Gerade g. Des Weiteren haben
123 -- wir noch den Eingabeparamter a,
124 -- welcher die Differenz zwischen
125 -- X2 und X1 darstellt. Aus den 3
126 -- Eingabeparametern soll die
127 -- Funktion den Wert y so bestimmen,
128 -- dass die Steigungen beider Geraden
129 -- gleich sind. Man stellt die
130 -- Gleichung nun so um, dass die
131 -- Funktion parallelThroughX den
132 -- Wert y berechnet.
133
134 crossingAt :: Float -> Float -> Float -> Float -> Float
135 crossingAt xg yg xh yh
136   | yg / (-xg) /= yh / (-xh) = (yh - yg) / (yg/(-xg) - yh/(-xh)) -- Die Steigung
137   -- mg und mh sollen nicht
138   -- gleich sein. Man setzt
139   -- beide Geradengleichungen
140   -- gleich, und stellt die
141   -- Gleichung nach dem
142   -- gesuchten x-Wert um, der
143   -- dann die x-Koordinate
144   -- ausgibt.
145   | otherwise = error "error" -- Sind die Steigungen
146   -- gleich, gibt die Funktion
147   -- error aus.
148

```