Aufgabe 1

a)
Es werden zu viele Kollisionen erzeugt, da:

$h(x) = h(x+1) = h(x+2) = ... = h(x+m-1)$

b)
Bei dieser Hashfunktion werden die geraden Hashslots nie befüllt, was natürlich schlecht ist, da nur die Hälfte der Hashtabelle benutzt werden kann.

c)
es sei $x \; = \; m-1$

$h(m-1) \; = \; m-1 \mod m + \lfloor \frac{m}{m-1+1} \rfloor \; = \; m-1+1 \; = \; m$

Es gibt aber keinen Eintrag mit dem Hashindex m, die Tabelle geht nur bis m-1, es gibt also beim insert einen error.

d)
Das geht leider nicht, da wir sonst nicht mehr die find Funktion auf die Hashtabelle ausführen können, da wir einen anderen Hashwert beim "find" bekommen als beim "insert" in die Tabelle.

e)
Wenn man $x = 0$ einsetzt haben wir $\frac{M}{0}$ was natürlich nicht möglich ist, da teilen durch Null nicht definiert ist.

f)
Hier verwenden wir also gute Algorithmen, aber dadurch dass wir hier l Algorithmen hintereinander verwenden steigt unsere Time Complexity von $\mathcal{O}(1)$ auf $\mathcal{O}(l)$, was natürlich schlecht ist.

Aufgabe 2

a)
es sei $x = m$ und $y \;=\; x + m$ für a beliebig
$h(x) \;=\; a \cdot m^2 \mod m \;=\; 0$
$h(y) \;=\; a \cdot (m + m)^2 \mod m \;=\; 0$
$\Rightarrow \quad x \neq y \;, \quad h(x) = h(y)$

Der Hashwert von $h(y)$ ist also gleich dem Hashwert von $h(x)$, für alle $a \in \mathcal{S}$. Da alle Hashwerte kollidieren, ist die Menge $\mathcal{H}_1$ nicht c-universell.