

```

      *
      |
    * /-\ *
    | /---\ |
  * /-----\ *
  | /-----\ |
* /-----\ *
| /-----\ |

```

Wie das oben gezeigte Bild schon andeutet, ist dies der Weihnachtszettel, der - wie sich das gehört - auch ein kleines Geschenk enthält. Ganz freiwillig ist er leider nicht, denn es ist schon wichtig, sich mit dem Inhalt zu beschäftigen, aber diesmal werden alle gelösten Teilaufgaben bewertet, aber nur 8 Punkte auf die Semester-Sollpunktzahl angerechnet. Man bekommt also mit der Lösung von Teilaufgabe a) schon 62,5% für den Zettel und kann insgesamt 200% erreichen.

Aufgabe 1:**Weihnachtsbäume**

(5 + 5 + 3 + 3 Punkte)

Die Funktion `putStr` wird später in der Vorlesung noch genauer behandelt. Sie müssen für diese Übung nur wissen, dass sie als Eingabe einen String erhält und diesen dann auf dem Bildschirm darstellt, wobei das Newlinesymbol `\n` die Darstellung in die nächste Zeile überleitet. Außerdem muss man beachten, dass man zur Darstellung des `\` ein `\\` angeben muss. Die Spitze des oben dargestellten Weihnachtsbaums erhält man also durch:

```
putStr "      * \n      | \n    * /-\ * \n    | /---\ | \n  * /-----\ * \n  | /-----\ | \n* /-----\ * \n| /-----\ | \n"
```

Dabei müssen natürlich auch die linksstehenden Leerzeichen genau abgezählt werden. Die Aufgabe besteht nun darin, einen solchen Baum automatisch zu zeichnen, wobei als Eingabeparameter nur die Anzahl n der Kerzenreihen verwendet wird (in unserem Beispiel ist $n = 4$). Es gibt 5 Punkte für eine funktionierende Lösung, z.B. indem Sie die Anregungen aus a) umsetzen oder aber einen eigenen Ansatz erarbeiten. Zusätzliche Punkte gibt es, wenn man **darüber hinaus** eine Lösung mit Nutzung höherer Listenfunktionen nach den Vorgaben aus b) bis d) erarbeitet.

a) Der Baum mit n Kerzenreihen hat $2n$ Zeilen und die längste Zeile hat $4n - 1$ Zeichen. Man überlegt sich also zuerst, wie die k -te Zeile aussehen muss und schreibt eine Funktion zu ihrer Erzeugung (k gerade/ungerade beachten). Zur Hilfe sollte man eine Funktion definieren, die einen String aus i Kopien eines gegebenen Zeichens erzeugt. Man kann dann eine Liste der Zeilenstrings erzeugen und daraus den Ausgabestring ableiten.

b) Mit diesem Ansatz codieren wir die Zeilen zuerst als Liste von Typ `[Int]`, wobei negative Zahlen als Leerzeichen, Nullen als `*`, Einsen als `|` und Zahlen $j > 3$ als `-` interpretiert werden. Nur die 2 macht Probleme, denn Sie kann ein `/` oder ein `\` bedeuten.

Deshalb stellt man zuerst nur die linke Hälfte des Baums (ohne Mittelachse) dar und setzt in jede Zeile (und nicht nur in jede zweite) eine Kerze.

- Die unterste Zeile wird durch $[1..(n-1)]$ repräsentiert, eine Funktion `next :: [Int] -> [Int]` soll die jeweils nächste Zeile erzeugen.
- Konstruieren Sie mit einer Funktion `build` eine `[[Int]]` Liste, die alle Zeilen der Zeichnung repräsentiert. Überlegen Sie selbst, wie man eine Abbruchbedingung einbauen kann.
- Jetzt muss die Liste vom Typ `[[Int]]` in eine Liste vom Typ `[String]` umgewandelt werden. Welche Zahlen müssen in welche Zeichen umgewandelt werden? Verwenden Sie dazu die höhere Listenfunktion `map`.
- Die Liste der Strings muss nun nur noch in einen String mit Newlinesymbolen verwandelt werden. Auch wenn das etwas mühevoll ist, sollten Sie versuchen, diese Aufgabe mit einer Faltung zu lösen.

c) Wer es perfektionieren will, macht das Bild symmetrisch (Mittelachse und Spitze extra einfügen) und sorgt d) dafür, dass nur in jeder zweiten Zeile eine Kerze steht.