**Programmieren 2** **Übungszettel 3** **SS2010**

***Aufgabe 1) Backtracking***

Eine Maus sitzt in einem Labyrinth und hungert. Irgendwo in diesem Labyrinth liegt Futter. Das Futter ist mit ‘F‘ gekennzeichnet. Begehbare Wege sind durch ein Leerzeichen angezeigt, Wände durch ein ‘#‘. Schreiben Sie ein Programm, welches die Maus steuert. Die Maus darf sich nach oben, unten, rechts oder links bewegen, aber nur wenn auf dem Zielfeld keine Mauer steht. Haben Sie keinen Weg zum Futter gefunden, so geben Sie eine entsprechende Meldung aus. Haben Sie einen Weg gefunden, so geben Sie bitte das Labyrinth mit dem eingezeichneten Weg aus.
Benutzen Sie bitte das Prinzip des Backtracking mittels einer rekursiven Implementierung.
Hinweis: Merken sie sich die Stellen im Labyrinth, an denen Ihre Maus schon war.

Das Labyrinth sieht folgendermaßen aus:

```
####################
# # # # ###   # #  #
# # # # #   # # # ##
#   # # # ### # #  #
### # # #   # # ## #
#   # #   # # #    #
# ### # ##### # # ##
# #   # # # ### #  #
# # ###   #  #  ## #
#   # ### ## # #   #
### # # # #  ### # #
# # ###      # # # #
#   # # ###### # # #
# ###     #    # # #
# #   ### # ## ### #
#   # # # # #    # #
### # # ### # #### #
#     # # #        #
# #####  F  # #### #
####################
```

Eine vorbereitete C-Datei mit dem Labyrinth finden Sie auf K:\mea\SS2010\PG2\Uebung\maze.c

Eine mögliche Form der Ausgabe wäre z.B. die folgende (Maus startet auf dem freien Feld oben links an Koordinate [1,1]):

```
####################
#.# # # ###...# #  #
#.# # # #...#.# # ##
#...# # #.###.# #  #
###.# # #.  #...## #
#...# #...# # #... #
#.### #.##### # #.##
#.#   #.# # ### #..#
#.# ###...#  #  ##.#
#...# ###.## # #  .#
###.# # #.#  ### #.#
# #.###...   # # #.#
#...# #.###### # #.#
#.###...  #    # #.#
#.#...### # ## ###.#
#...# # # # #    #.#
### # # ### # ####.#
#     # # #........#
# #####  F..# #### #
####################
```

***Aufgabe 2) Aufzählungstypen***

Wir möchten das Verhalten von logischen Spannungspegeln auf einem elektrischen Bussystem simulieren. Dazu erlauben wir die folgenden Werte auf dem Modell des Busses:

- ‚1' entspricht einer logischen 1
- ‚0' entspricht einer logischen 0
- ‚X' entspricht einer einem undefinierten Zustand
- ‚H' entspricht einer „weichen" logischen 1 (Abschluss mit Pull-Up-Widerstand)
- ‚L' entspricht einer „weichen" logischen 0 (Abschluss mit Pull-Down-Widerstand)
- ‚Z' entspricht dem hochohmigen Zustand (kein treibendes Signal)

Bei der Simulation können nun mehrere Teilnehmer ein elektrisches Signal mit einem Wert aus den oben definierten Werten auf den Bus setzen. Der resultierende Zustand des Busses wird dann gemäß der folgenden Tabelle für die Zusammenschaltung von zwei Teilnehmern berechnet:

| | 1 | 0 | X | H | L | Z |
|---|---|---|---|---|---|---|
| 1 | 1 | X | X | 1 | 1 | 1 |
| 0 | X | 0 | X | 0 | 0 | 0 |
| X | X | X | X | X | X | X |
| H | 1 | 0 | X | H | X | H |
| L | 1 | 0 | X | X | L | L |
| Z | 1 | 0 | X | H | L | Z |

a) Deklarieren Sie einen Aufzählungstypen `enum LogicLevel`, der obige Werte enthält. Schreiben Sie anschließend eine Funktion, die die Berechnung des Ergebnisses der Verschaltung von zwei solchen Werten berechnet:

```
LogicLevel resolve(LogicLevel p1, LogicLevel p2);
```

Benutzen Sie zur Auswertung und Berechnung bitte die **`switch`**-Fallunterscheidungen.

b) Definieren Sie eine Ihnen schlüssig erscheinende Tabelle, die ein logisches AND durchführt, bei dem immer 0 oder 1 zurückgeliefert wird. Implementieren Sie diese Funktion:

```
LogicLevel AND(LogicLevel p1, LogicLevel p2);
```

c) Zwecks Test der Funktionen schreiben Sie eine kleine Simulation von 4 Schritten für das folgende System: Fa und Fb produzieren ‚1', ‚0' oder ‚Z' und beide hängen am gemeinsamen Bus, der mit einem Pull-Up versehen ist. Das AND wird auf dem einen Eingang mit einer ‚1' versorgt, der andere Eingang hängt am Bus. Führen Sie in der Simulation die folgenden 4 Schritte durch und geben Sie in jedem Schritt *bus* und *out* aus:

   a. Fa und Fb beide hochohmig (auf ‚Z')
   b. Fa liefert nun ‚0'
   c. Fb schaltet auf ‚1'
   d. Fa schaltet hochohmig ‚Z'