3. Übungsaufgabe zu
Fortgeschrittene Funktionale Programmierung
Thema: Memoration vs. Dynamische Programmierung
ausgegeben: 11.05.2005, fällig: 18.05.2005

Für dieses Aufgabenblatt sollen Sie Haskell-Rechenvorschriften zur Lösung der im folgenden angegebenen Aufgabenstellungen entwickeln und für die Abgabe in einer Datei namens `AufgabeFFP3.hs` in Ihrem Homeverzeichnis ablegen, wie gewohnt auf oberstem Niveau. Kommentieren Sie Ihre Programme aussagekräftig und benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten.

- Nach dem erfolgreichen Abschluss des Projekts *The Gates* im Central Park von New York tragen sich Christo und Jeanne-Claude nun mit der Idee, die schachbrettartig angelegten Straßen von Manhattan mit Stoffbahnen zu verhüllen. Diese sollen aus Abschnitten unterschiedlicher Länge und Farbe zusammengenäht sein, wobei als Stoffabschnitte ausschließlich Abschnitte von einem, zwei und drei Yard Länge in den Farben lindgrün und türkisblau verwendet werden sollen. Da die optische Gesamtwirkung entscheidend von der Komposition der Stoffbahnen abhängt, interessieren sich Christo und Jeanne-Claude zum jetzigen Zeitpunkt besonders für die Anzahl von Stoffbahnen, aus denen sie wählen können, eine Straße von n Yard Länge zu verhüllen.

  Tragen Sie zum Erfolg dieses neuen Projekts bei, indem Sie folgende Haskell-Rechenvorschriften schreiben:

  - eine Haskell-Rechenvorschrift `anzEinfaerbig :: Integer -> Integer`, die, angewendet auf ein Argument n, die die Länge der Straße in (ganzen) Yard angibt, ausgibt, aus wieviel einfärbigen Stoffbahnen zur Verhüllung dieser Straße ausgewählt werden kann,
  - eine Haskell-Rechenvorschrift `anzZweifaerbig :: Integer -> Integer`, die angibt, wieviel zweifärbige Stoffbahnen unter Christos und Jeanne-Claudes Vorgaben hinsichtlich Länge und Farbe für diese Verhüllung geschneidert werden können.

- Überlegen Sie sich, ob Sie durch Memoration und dynamische Programmierung zu effizienteren Implementierungen der Funktionen `anzEinfaerbig` und `anzZweifaerbig` kommen können. Wenn ja, geben Sie Ihre Implementierungen der gedächtnisbehafteten Varianten

- von `memo_anzEinfaerbig :: Integer -> Integer` und `memo_anzZweifaerbig :: Integer -> Integer`, bzw.
- von `dyn_anzEinfaerbig :: Integer -> Integer` und `dyn_anzZweifaerbig :: Integer -> Integer`

an.