1. Die Zahlen von 1 bis 10 sollen

   (a) absteigend mit einer `do while`-Schleife

   (b) aufsteigend mit einer `do while`-Schleife

   (c) absteigend mit einer `while`-Schleife

   (d) aufsteigend mit einer `while`-Schleife

   ausgegeben werden. Es ist dabei genau zu beachten, ob ein Jump-Befehl am Anfang oder am Ende der Schleife erfolgt.

2. Die Zahlen von 1 bis zu einer einzulesenden Zahl sollen aufaddiert und die Summe ausgegeben werden.

3. Die Struktur von `if else` ist in möglichst allgemeiner Form in eine Kombination von Jump-Befehlen zu übersetzen.

4. Eine Zahl wird eingelesen und festgestellt, ob sie gerade oder ungerade ist. Bei einer geraden Zahl soll 0, sonst 1 ausgegeben werden. Die Ermittlung des Restes bei der Division durch 2 soll in einem Unterprogramm gemacht werden.

5. Zwei Zahlen werden eingelesen und die größere der beiden Zahlen wird ausgegeben.

6. Ein Programm soll zwei positive Zahlen einlesen, in einem Unterprogramm `mult` das Produkt berechnen und das Ergebnis ausgeben. Zuerst ist dazu ein entsprechendes Java-Programm zu schreiben, wobei darin Multiplikation nicht erlaubt ist. Dieses Java-Programm ist dann in ein Maschinen-Programm „zu Fuß zu kompilieren“.

   Schwierigere Variante: Die beiden Zahlen können negativ oder nicht-negativ sein.

7. Die Multiplikation kann auch rekursiv über Additionen erfolgen. Schreibe dazu ein Java-Programm ohne Parameterübergabe und entwirf mit Hilfe dieses Programmes ein DCL-Maschinen-Programm.

   Warum kann diese rekursive Variante der Multiplikation nicht alle im DC erlaubten Faktoren verarbeiten?

   Tipp: Mit `v Speicheradresse` (v steht für *view*) kann man sich den Stack im DC ansehen.