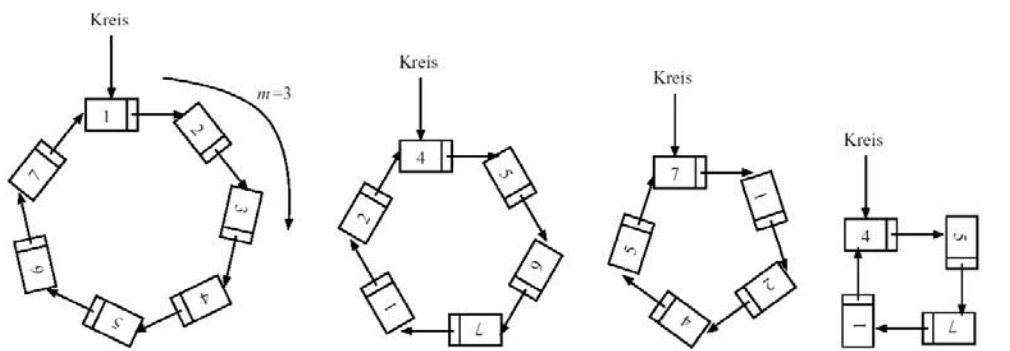


**1) Das Josephus-Problem**

Der jüdische Historiker Josephus Flavius (37 - 95) berichtete, dass er zusammen mit 40 anderen Juden in einen Keller geflüchtet sei, als die Römer eine jüdische Stadt eroberten. Um dem Feind nicht in die Hände zu fallen, wollten sie sich gegenseitig selbst umbringen. Josephus war dagegen, konnte die anderen jedoch nicht umstimmen. Um dem Tod zu entgehen, schlug er vor, man solle sich im Kreis aufstellen und abzählen. Jeder Dritte solle auf der Stelle umgebracht werden. Sein Vorschlag wurde akzeptiert. Josephus stellte sich so in den Kreis, dass er als letzter übrig und damit am Leben blieb.

Das Problem des Josephus lässt sich mit einfach verketteten Ringlisten simulieren. Schreiben Sie ein Programm, das die Reihenfolge auflistet, in der die Personen gewählt werden. Das Programm bekommt als Eingabe die Zahl  $n$  der Personen am Anfang im Kreis und die Zahl  $m$ , um die weitergezählt wird.

Die folgende Grafik zeigt am Beispiel  $n = 7, m = 3$ , wie sich die Datenstruktur entwickeln soll:



Ausgabe:      3                                  6                                  2                                  7                                  5 1 Josephus: 4

- a) Entwerfen Sie eine geeignete Datenstruktur `Person`.
- b) Schreiben Sie eine Prozedur `Person* build(int n)` welche die Datenstruktur aufbaut: Der zurückgegebene Zeiger zeigt auf die erste der  $n$  Personen im Kreis.
- c) Schreiben Sie ein Prozedur `void josephus(Person *&ring, int m)` die die Liste in der beschriebenen Art abbaut: Es soll solange jedes  $m$ -te Element aus der Liste `ring` entfernt werden, bis nur noch eines übrig ist. Jedemal, wenn ein Element aus der Liste entfernt wird, wird dessen Nummer ausgegeben.
- d) Vervollständigen Sie alle Teile zu einem lauffähigen Programm, das nach Eingabe von  $n$  und  $m$  die ermittelte Reihenfolge der Personen ausgibt.

Abgabetermin: 14. Januar 2004