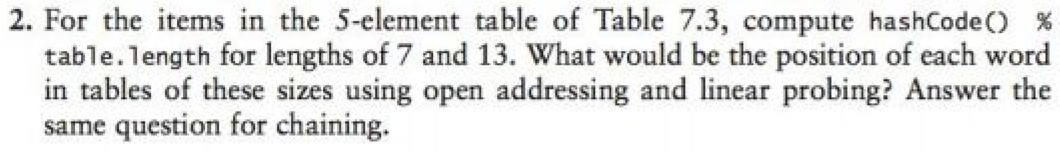
Self-Check2



2. For the items in the 5-element table of Table 7.3, compute `hashCode() % table.length` for lengths of 7 and 13. What would be the position of each word in tables of these sizes using open addressing and linear probing? Answer the same question for chaining.

Name	hashCode	hashCode % 7	hashCode % 13
“Tom”	84274	1	8
“Dick”	2129869	0	1
“Harry”	69496448	0	8
“Sam”	82879	6	4
“Pete”	2484038	4	11

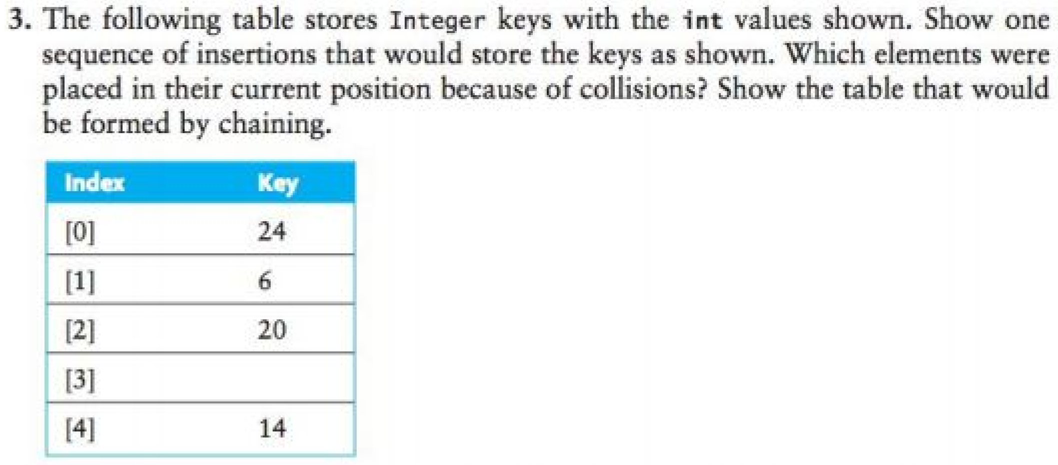
hashCode % 7

0	1	2	3	4	5	6
“Dick”	“Tom”	“Harry”	null	“Pete”	null	“Sam”

hashCode % 13

0	1	2	3	4	5	6	7	8	9	10	11	12
null	“Dick”	null	null	“Sam”	null	null	null	“Tom”	“Harry”	null	“Pete”	null

Self-Check3



3. The following table stores `Integer` keys with the `int` values shown. Show one sequence of insertions that would store the keys as shown. Which elements were placed in their current position because of collisions? Show the table that would be formed by chaining.

Index	Key
[0]	24
[1]	6
[2]	20
[3]	
[4]	14

hashKey() % 5
insert order: 14, 24, 6, 20

operations:
14 % 5 = 4 add 14 to [4] index
24 % 5 = 4 add 24 to [4] index - collision - move to [0]
6 % 5 = 1 add 6 to [1] index
20 % 5 = 0 add 20 to [0] index collision - add 20 to [1] index - collision - add 20 to [2] index