TECHNICAL UNIVERSITY OF APPLIED SCIENCES ULM

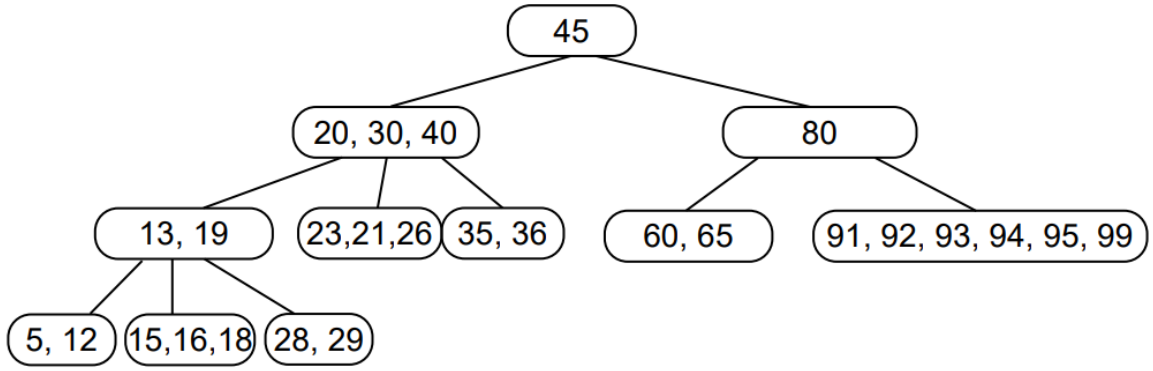
Assignment #5

Student names and Group: *Petar Stoyanov & Eetu Hyvärinen - Group 08*

Course: *Algorithms and Data Structures* – Professor: *Prof. Schied*
Due date: *January 9th, 2020*

Exercise 5.2

The following tree should be a B-tree of order **m=3**.
Identify all the errors contained in it and indicate what is wrong in each case.

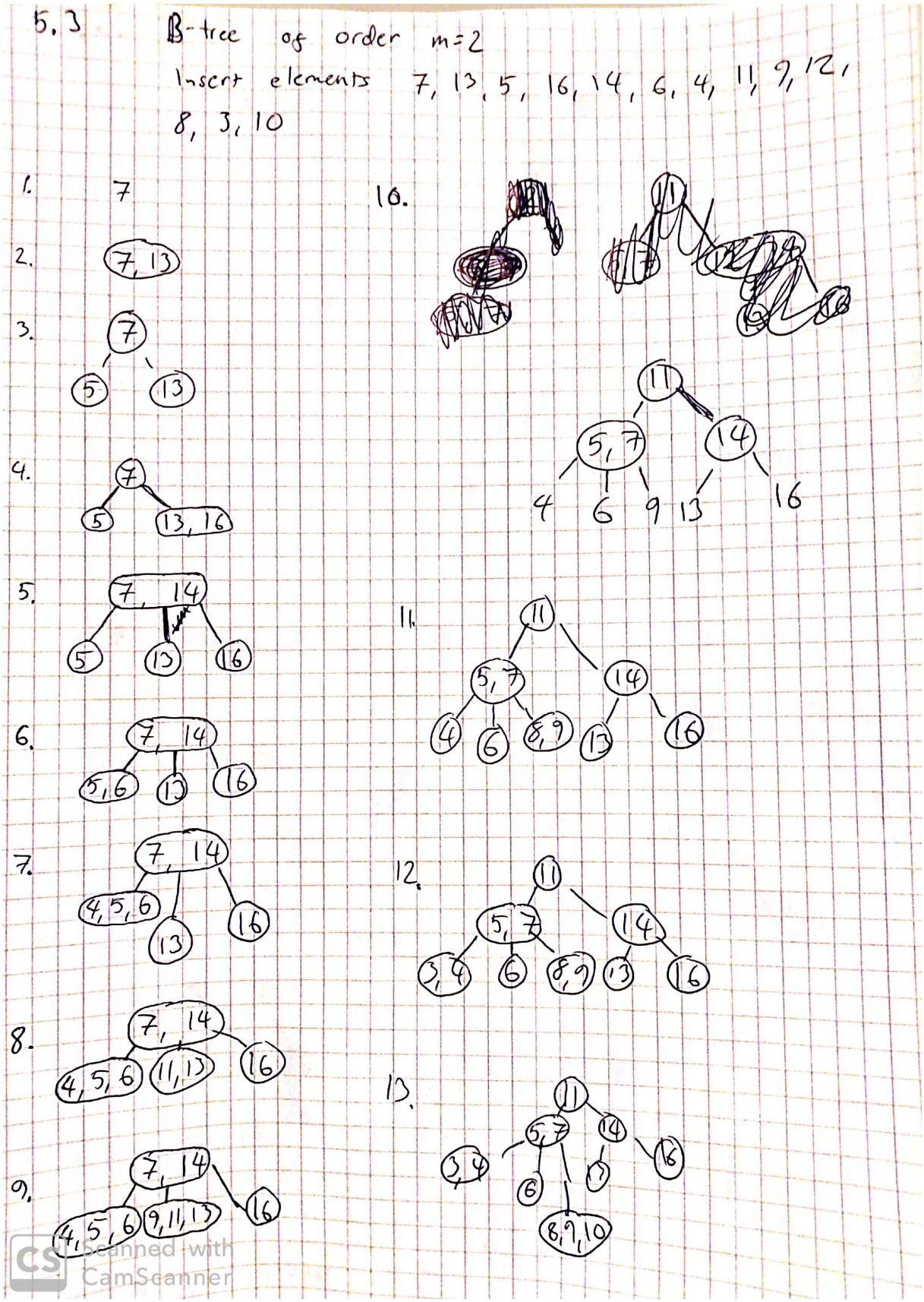


1. All leaves should be on the same level. (5,12) , (15,16,18) and (28,29) are too low.
2. A non leaf node with k children should contain k-1 keys. (20, 30, 40) has too many keys, since m=3
3. All nodes (including root) may contain at most 2m – 1 keys. (80's right side has 6, should be a maximum of 5)
4. (35, 36) are not greater than 40, therefore they shouldn't be on the right side of 40.
5. (28,29) can't be on the left side of 20 and on the right side of 19
6. Leaf nodes can have a maximum of m-1 keys

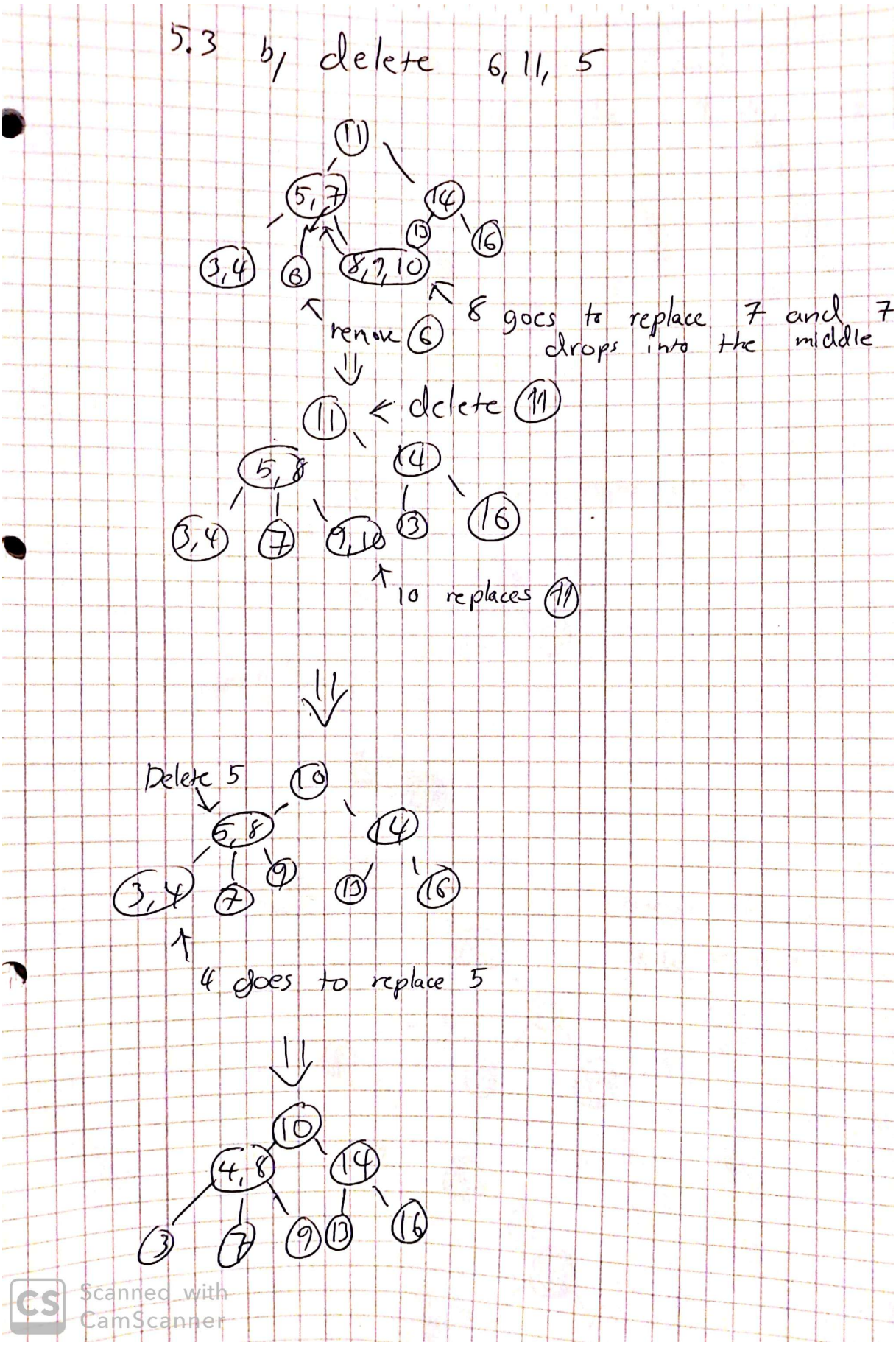
Exercise 5.3

1. Insert the elements
   7, 13, 5, 16, 14, 6, 4, 11, 9, 12, 8, 3, 10
   successively into a B-tree of order **m=2** (a so-called 2-3-4 tree) that is empty at the beginning. Which B-tree results?
2. Then delete the elements 6,11 and 5 from the tree. Specify the essential intermediate steps so that the method of solution is visible.

(*a*)

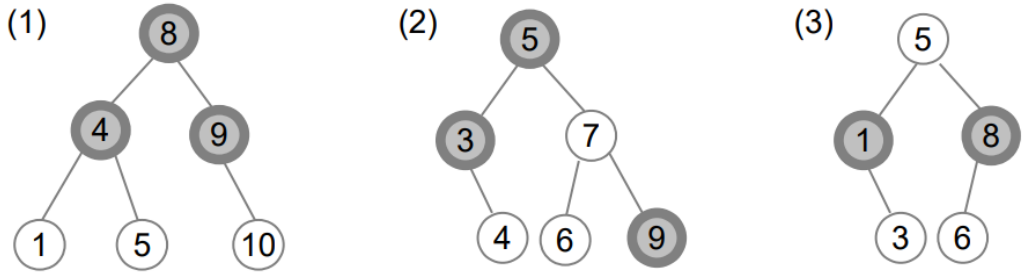

5.3
B-tree of order m=2
Insert elements 7, 13, 5, 16, 14, 6, 4, 11, 9, 12, 8, 3, 10
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
Scanned with CamScanner

(*b*)

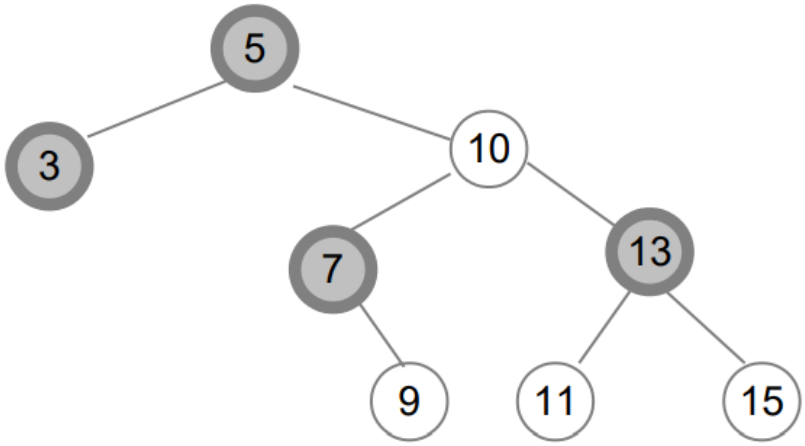


Exercise 5.4

1. Which of the following trees are red-black trees? Give a reason if not (black = thick border and filled with grey; red = thin border; nil nodes are not shown).

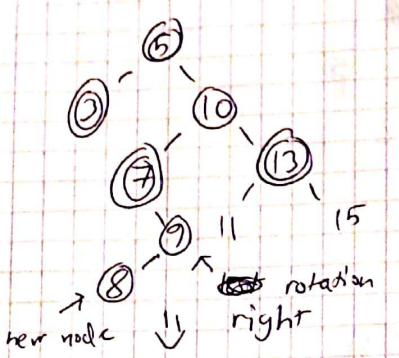


2. Insert the values 8 and 12 successively into the following red-black tree

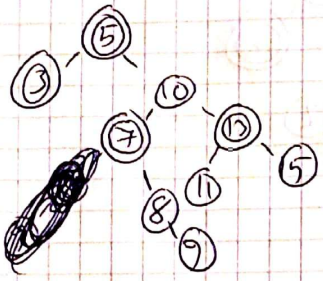


1. Is a red-black tree, as there is an equal amount of black nodes from any given node to any of its descendant NIL nodes
2. Is not a red-black tree, as when a node is red, both of its children have to be black, but 6 is a red node although its coming from another red node.
3. According to properties of red-black trees, the root node has to be black. So although it has little impact on analysis of the tree, this is not a red-black tree.

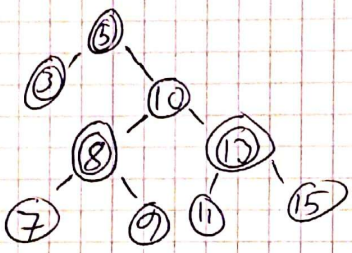
5.4 b) insert 8



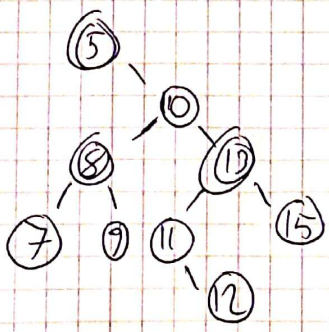
Step 1 right rotation on 9



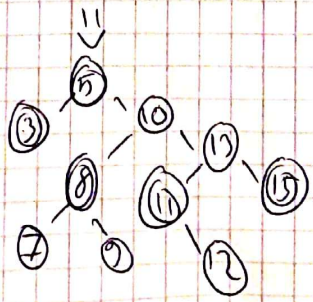
Step 2: left rotation on 7



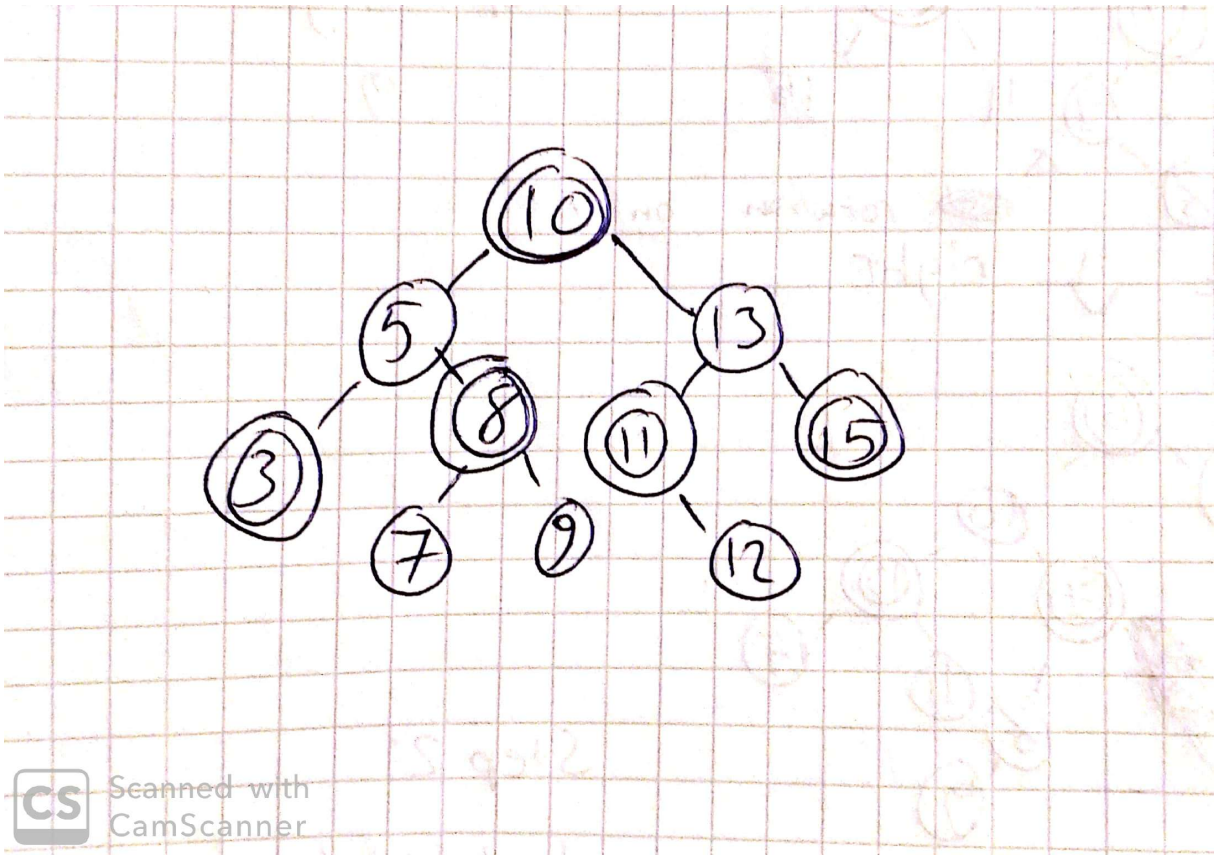
Insert 12



recolour 11 and 15 black and 13 red



left rotation on 5 and recolor on 10 and 5



Exercise 5.5

We use a hash table to store the favorite colors of persons and the hash function h for character strings s = $c_1c_2c_3$.... c_j is used :

(s) = ((pos (c1) + pos (c2)) % m,

where m is the size of the hash table and

pos: {A, B,..., Z} → {1, 2, . . . , 26}

assigns the position in the alphabet to each character from A to Z (i. e. the alphabet positions of the first two characters of the string are added modulo m).

c	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
pos(*c*)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

The following data is now entered one after the other into a hash table of **size m=10**. Collision resolution is done using **chaining**

EMMA	blue
BEN	yellow
MIA	purple
PAUL	green
HANNAH	red
LEON	black

EMILIA	pink
FINN	neon
SOPHIA	red
ELIAS	white

What is the final state for the hash table?

Name	**(c1 + c2)**	**mod 10**
EMMA	(5 + 13)	= 8
BEN	(2 + 5)	= 7
MIA	(13 + 9)	= 2
PAUL	(16 + 1)	= 7
HANNAH	(8 + 1)	= 9
LEON	(12 + 5)	= 7
EMILIA	(5 + 13)	= 8
FINN	(6 + 9)	= 5
SOPHIA	(19 + 15)	= 4
ELIAS	(5 + 12)	= 7

HASHTABLE				
0	null			
1	null			
2	"MIA"purple			
3	null			
4	"SOPHIA"red			
5	"FINN"neon			
6	null			
7	"BEN" yellow →	"PAUL" green →	"LEON" black →	"ELIAS" white
8	"EMMA" blue →	"EMILIA" pink		
9	"HANNAH" red			