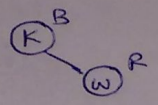
Assignment - 5

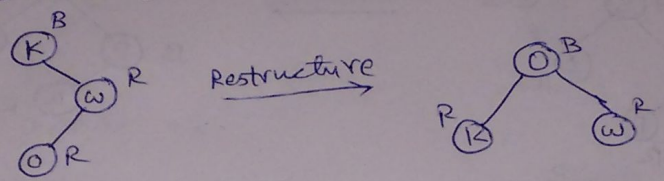
1 ans:

Step 1: Insert K

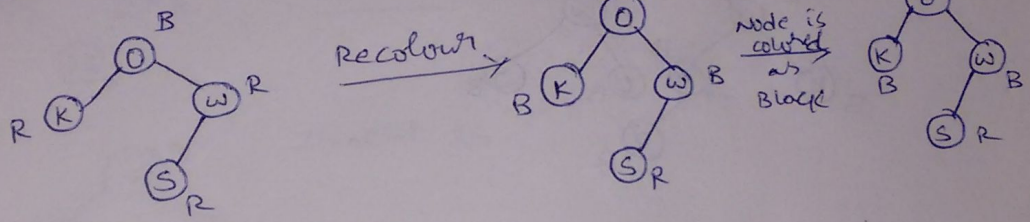
Step 2: Insert W



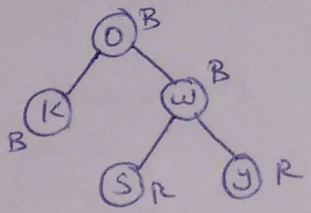
Step 3: Insert O



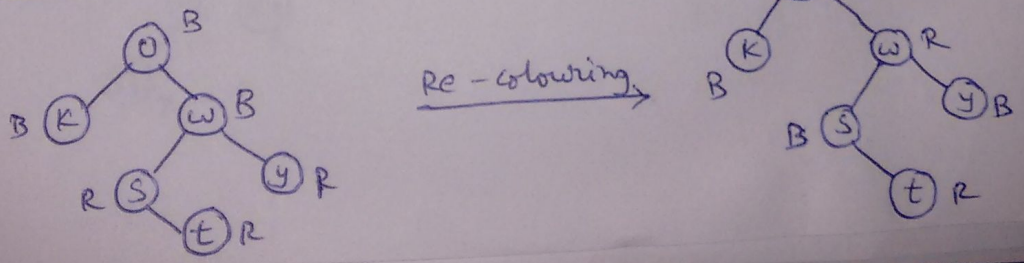
Step 4: Insert S



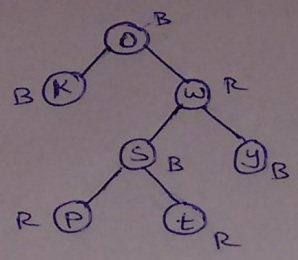
Step 5: Insert Y



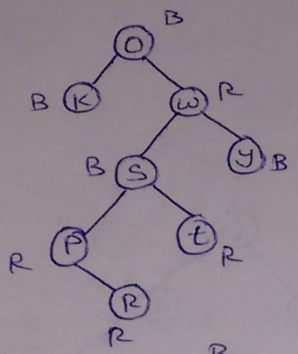
Step 6: Insert t



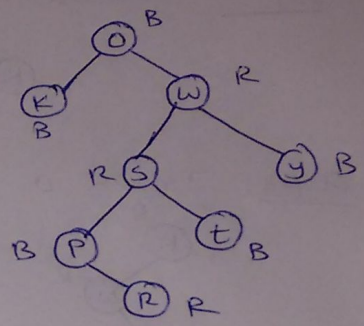
Step 7:



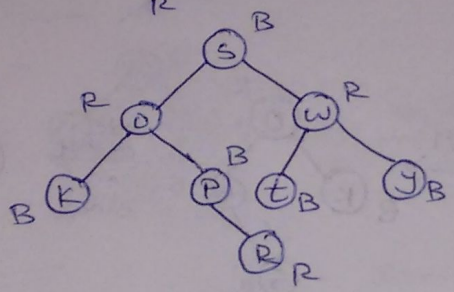
Step 8:



Recolour →



Restructure →



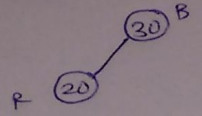
4 ans: (4)

$h(x) = x \bmod 13$

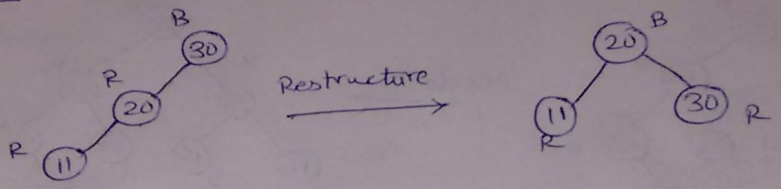
2 ans: (2)

Step 1: Insert 30

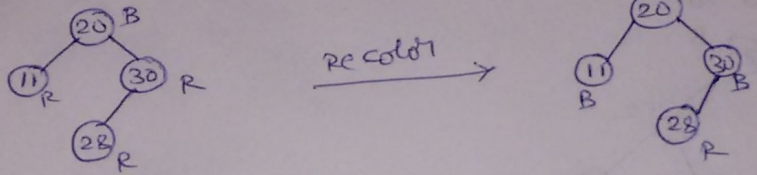
Step 2: Insert 20



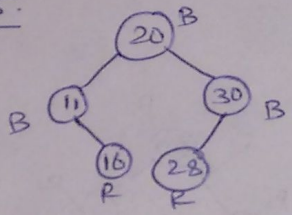
Step 3: Insert 11



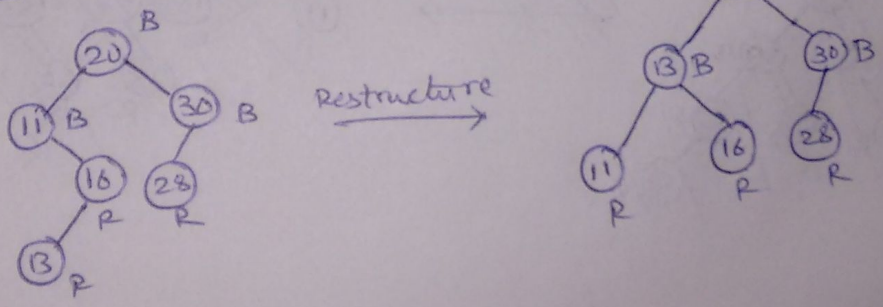
Step 4: Insert 28



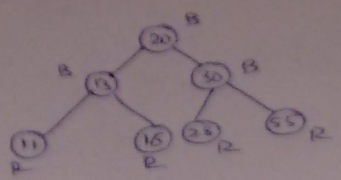
Step 5: Insert 16



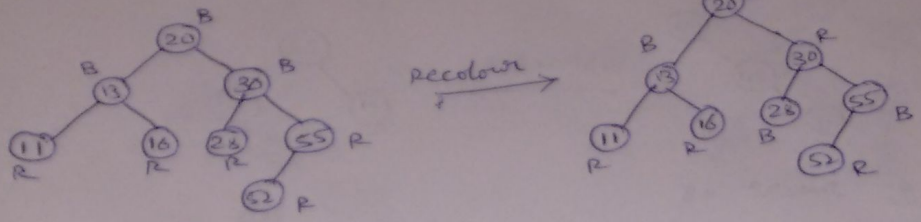
Step 6: Insert 13



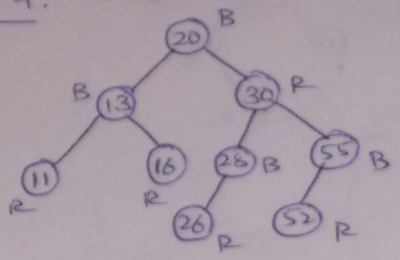
Step 7: Insert 55



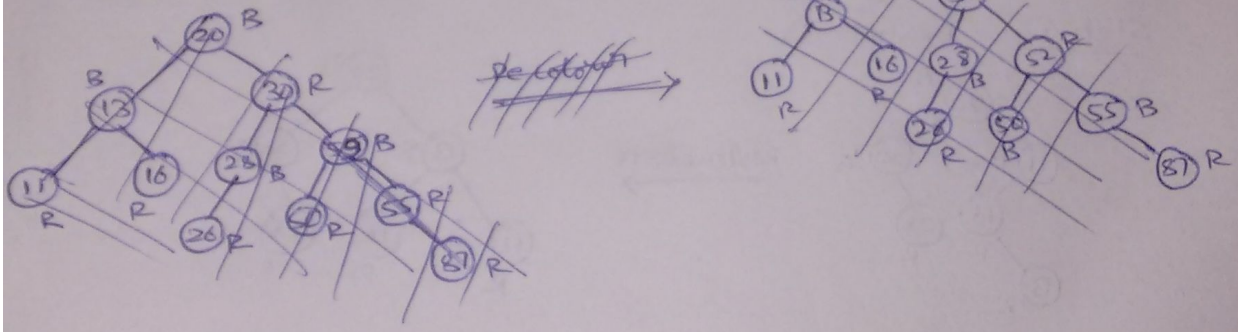
Step 8: Insert 52



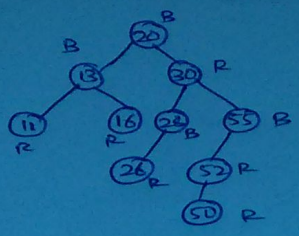
Step 9: Insert 26



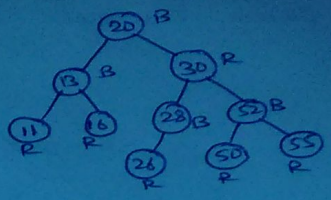
Step 10: Insert 50



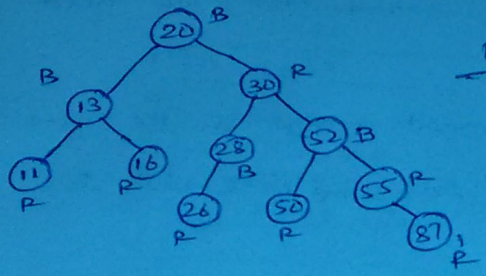
Step 10: Insert 50



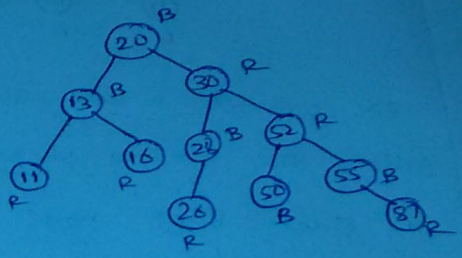
restructure →



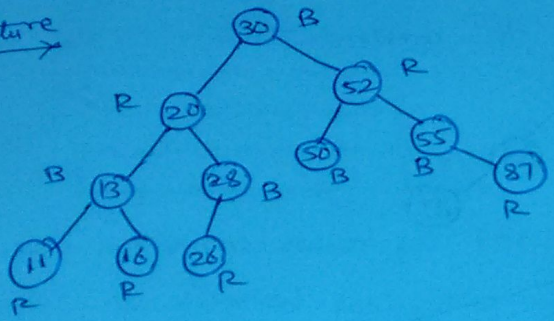
Step 11: Insert 87



Re color →



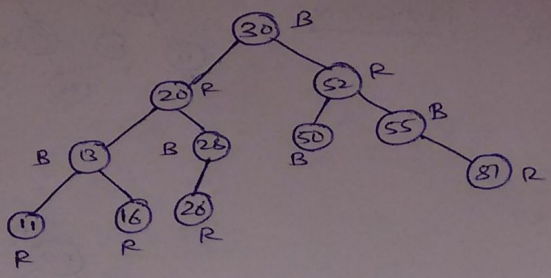
restructure →



4 ans:

3 ans:

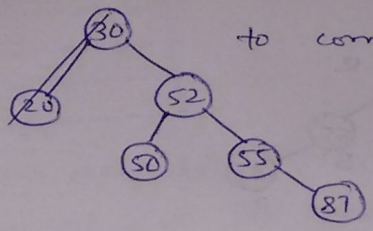
Given Red Black tree,



To convert a Red-Black tree to 2-3-4 tree, we need to follow the below steps:

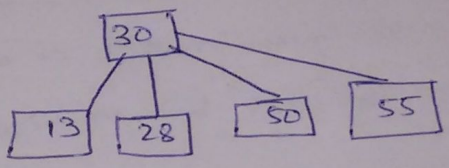
1) Each Black node should represent a node in the 2-3-4 tree
2) Each Red node should be in the same node with its black parent.

restructure →

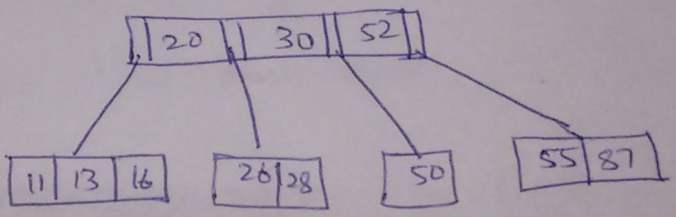


to convert this:

Step 1:



Step 2:



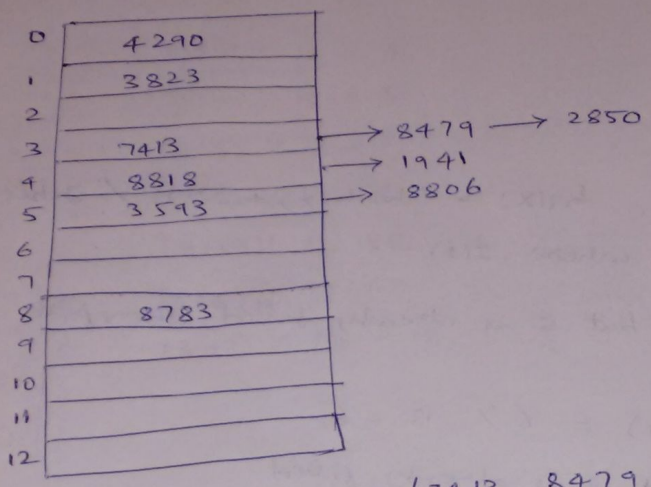
4 ans:

$h(x) = x \bmod 13$.

So,

$3823 \,\%\, 13 = 1$
$8806 \,\%\, 13 = 5$
$8783 \,\%\, 13 = 8$
$2850 \,\%\, 13 = 3$
$3593 \,\%\, 13 = 5$
$8479 \,\%\, 13 = 3$
$1941 \,\%\, 13 = 4$
$4290 \,\%\, 13 = 0$
$8818 \,\%\, 13 = 4$
$7413 \,\%\, 13 = 3$

Hash Table:



Here the collisions occur for (7413, 8479, 2850). So we need to create a linked list. The new element is at the starting of linked list. So as shown above 7413 is in first position, 8479 second position and 2850 at last position. The same is the case for collision of (8818, 1941) and (3593, 8806)

So, the final Hash table is shown above.

Sol:

Table size = 13

Hash function $h(X) = X \bmod 13$

0	4290
1	3823
2	
3	2850
4	8479
5	8806
6	3593
7	1941
8	8783
9	8818
10	7413
11	
12	

In Linear Probing. $h_i(X) = (hash(x) + f(i)) \% \text{Table size}$

with $f(0) = 0$ where $f(i) = i$

$3593 \% 13 = 5$, but 5 is already filled, therefore Linear probing 3593

$(5+1) = 6 \% 13 = 6$

8479 at 3, But 3 is already filled

$(3+1) = 4 \% 13 = 4$

1941 at 4, which is filled

$(4+1) = 5 \% 13 = 5$

5 is also filled $= (4+2) = 6 \% 13 = 6$

$= (4+3) = 7 \% 13 = 7$

8818 at 4,

$(4+1)\%\ 13 = 5$

$(4+2)\%\ 13 = 6$

$(4+3)\%\ 13 = 7$

$(4+4)\%\ 13 = 8$

$(4+5)\%\ 13 = 9$

74 13 is at 3, But collision exists. Hence it is inserted at position 10. but continously checking

$(3+1)\%\ 13 = 4$

$(3+2)\%\ 13 = 5$

$(3+3)\%\ 13 = 6$

$(3+4)\%\ 13 = 7$

$(3+5)\%\ 13 = 8$

$(3+6)\%\ 13 = 9$

$(3+7)\%\ 13 = 10$

Ques

Table size = 13

$h_i(x) = (hash(x) + f(i)) \bmod tablesize$

where $f(i) = i^2$

0	1941
1	3823
2	
3	2850
4	8479
5	8806
6	3593
7	8818
8	8783
9	4290
10	
11	
12	7413

3593 should be inserted at 5, But collision is occuring.

$(5+1^2) = 6 \% 13 = 6$, so 6 is new position

→ 8479 should be inserted at position 3,

But, collision is present

$\therefore (3+1^2) = 4 \% 13 = 4$

→ 1941 is at position 4,

But there is collision

$4 + 1^2 = 5 \% 13 = 5$

$4 + 2^2 = 8 \% 13 = 8$

$4 + 3^2 = 13 \% 13 = 0$

→ 4290 at 0

$0 + 1^2 = 1 \% 13 = 1$

$0 + 2^2 = 4 \% 13 = 4$

$0 + 3^2 = 9 \% 13 = 9$

→ 8818 at 4,

$4 + 1^2 = 5 \% 13 = 5$

$4 + 2^2 = 8 \% 13 = 8$

$4 + 3^2 = 13 \% 13 = 0$

$4 + 4^2 = 20 \% 13 = 7$

→ 7413 at 3

$3 + 1^2 = 4 \% 13 = 4$

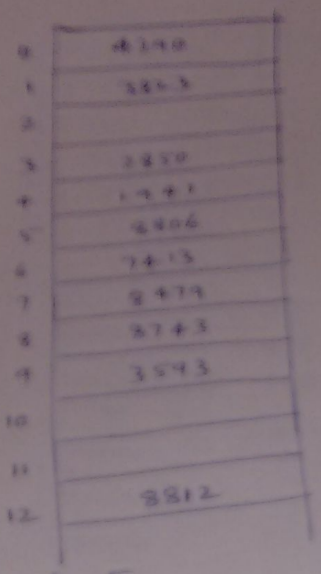
$3 + 2^2 = 7 \% 13 = 7$

$3 + 3^2 = 12 \% 13 = 12$

Ans:

$H_1(x) = x \bmod 13$

$H_2(x) = 11 - (x \bmod 11)$



3543 at 5,

$H_2 = 11 - (3543 \% 11)$

$= 11 - 7 = 4$

$5+4 = 9 \% 13 = 9$

8479 at 3,

$H_2(x) = 11 - (8479 \% 11)$

$= 11 - 9 = 2$

$3+2 = 5 \% 13 = 5$ occupied

$(5+2) \% 13 = 7$

8818 at 4, → occupied

$H_2(x) = 11 - (8818 \bmod 11) = 11 - 7 = 4$

$(4+4) \% 13 = 8$

$8+4 = 12 \% 13 = 12$.

7413 at 3 →

$H_2(x) = 11 - (7413 \bmod 11) = 11 - 10 = 1$

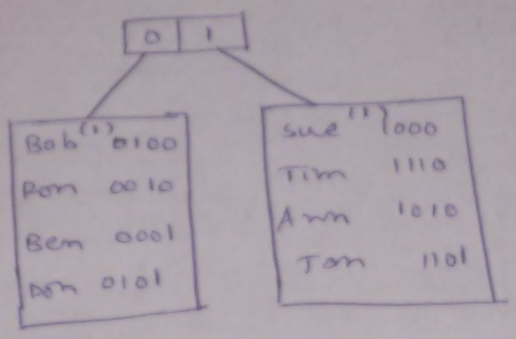
(3+1) % 13 = 4

(4+1) % 13 = 5

(5+1) % 13 = 6

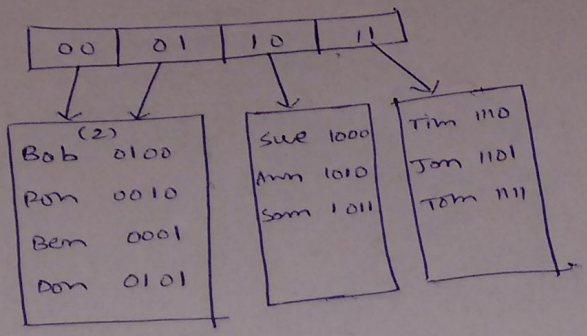
Bonus: Extensible Hash:

Bob	0100
Sue	1000
Tim	1110
Ron	0010
Ann	1010
Jon	1101
Ben	0001
Don	0101
Tom	1111
Sam	1011



Inserting Tom 1111 causes an overflow and the tables will be split

After Split, Sam 1011 is inserted



9 and 3:

Table 1

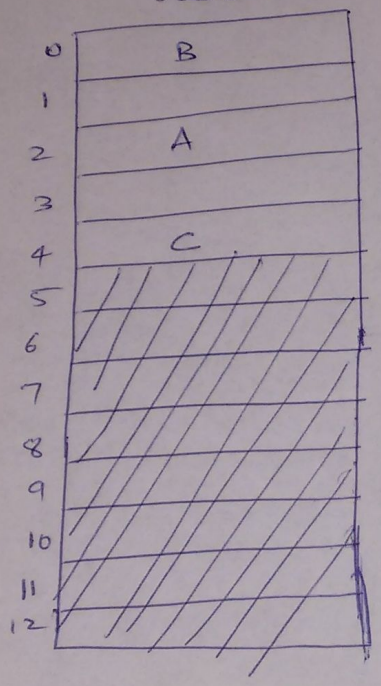
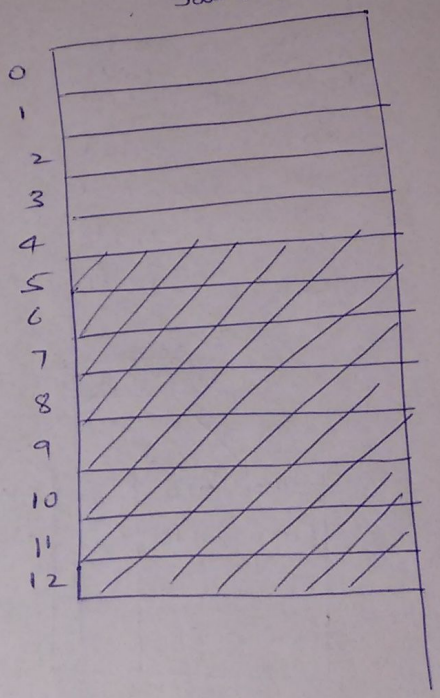


Table 2



D should be at 0 of table 1. But B is already there so it goes to 0 of table 2

Table 1

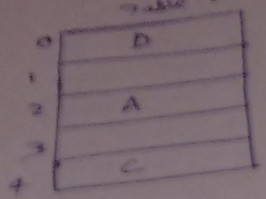
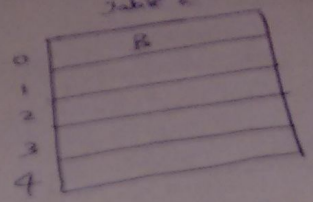


Table 2



E should be at place 2, But 2 is occupied by A

A should be shifted to position 0 of table 2

Table 1

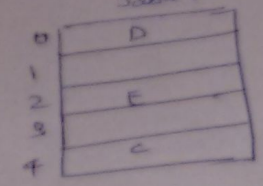
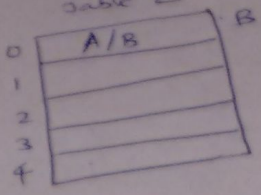


Table 2



Now B is shifted to 0 of Table 1

Table 1

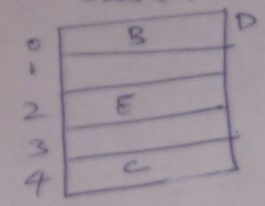
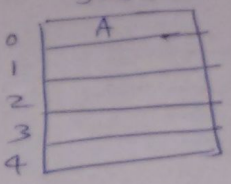


Table 2



D is shifted to position 1 of Table 2

Table 1

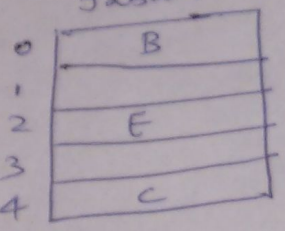
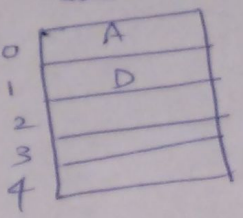


Table 2



10 ans

	Item	Hop
0		
1		
2		
3		
4		
5		1000
6	A	1000
7	B	
8		1000
9	C	
10		
11		
12		

D should be at 7, but already occupied (collision) so we need to see the next position.

0		
1		
2		
3		
4		
5		
6	A	1000
7	B	1100
8	D	0000
9	C	1000
10		
11		
12		

E should be inserted at 6, But 6 is occupied therefore B should be shifted to 10

0		
1		
2		
3		
4		
5		
6	A	1100
7	E	0101
8	D	0000
9	C	1000
10	B	0000
11		
12		

F is at 7, But 7 is occupied by E.
So C can be shifted "

0		
1		
2		
3		
4		
5		
6	A	1100
7	E	0111
8	D	0000
9	F	0010
10	B	0000
11	C	0000
12		

G should be inserted at 8, But D is present there, so C should be shifted to 12, so the table will be as below

0		
1		
2		
3		
4		
5		
6	A	1100
7	E	0111
8	D	0001
9	F	0001
10	B	
11	G	
12	C	1000

This is the Final Hash Table.