Name- Tanmay Mahakar
PRN- F22111067
SE comp I

DSA Assignment

Q1. Write a C/C++ function to print tree on DFS (without using recursion).

Ans.

```
void non_rec (node *T)
{
        stack s;
        while (T!=NULL)
        {
                s.push (T);
                T= T->Left;
        }
        while (!s.empty())
        {
                T = s.pop();
                cout <<T->data << " ";
                T=T->right;
                while (T!=NULL)
                {
                        s.push (T);
                        T =T->Left;
                }
        }
}
```

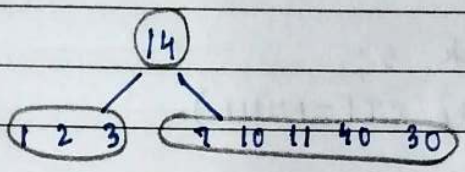
Q2. Construct a binary tree from given two traversals.

Inorder – 1 2 3 14 7 10 11 40 30

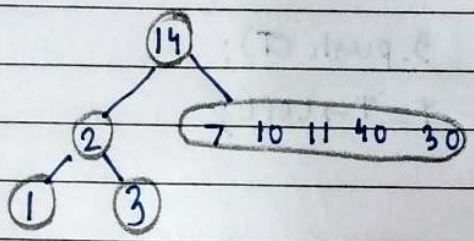
Postorder – 1 3 2 7 10 40 30 11 14

Ans.

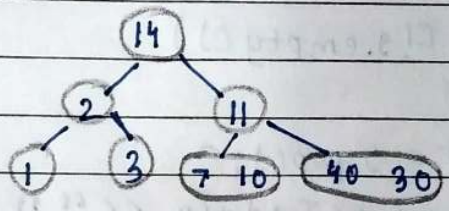
1)



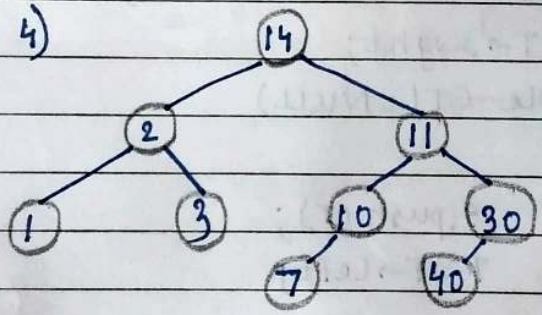
2)



3)



4)



Q3. Write a short note on hash functions.

Ans. The position of a key in a container (array) which is used to hold some number of elements of a given set K is given by a function h(.), known as the hash function.

- A good hash function should:-

i) Minimize collisions

ii) Be easy and quick to compute.

iii) Distribute key values evenly on the hash table

iv) Use all the information provided in the key.

. Methods of Hashing functions are:-

1) Division method
2) Multiplication method
3) Digit extraction method
4) Folding
5) mod-square method

Q4. For the given set of values.
11, 33, 20, 88, 79, 98, 44, 68, 66, 22
Create a hash table with size 10 and resolve collision using chaining with replacement and without replacement. Use the modulus hash function (key % size).

Ans. i) without replacement

i) • 11 →
11 mod 10 = 1

• 33 →
33 mod 10 = 3

• 20 →
20 mod 10 = 0

• 88 →
88 mod 10 = 8

• 79 →
79 mod 10 = 9

	Key	Chain
0	20	-1
1	11	-1
2		-1
3	33	-1
4		-1
5		-1
6		-1
7		-1
8	88	-1
9	79	-1

ii) • 98 →
98 mod 10 = 8 ∴ collision occurred
∴ new position = 2

• 44 →
44 mod 10 = 4

• 68 →
68 mod 10 = 8 ∴ collision occurred
∴ new position = 5

• 66 →

66 mod10 = 6

• 22 →

22 mod10 = 2 ∴ collision occured

∴ new position = 7

	key	chain
0	20	-1
1	11	-1
2	98	5
3	33	-1
4	44	-1
5	68	-1
6	66	-1
7	22	-1
8	88	2
9	79	-1

2) with replacement

i) • 11 → 11 mod10 = 1

• 33 → 33 mod10 = 3

• 20 → 20 mod10 = 0

• 88 → 88 mod10 = 8

• 79 → 79 mod10 = 9

ii) • 98 → 98 mod10 = 8 ∴ collision

∴ new position = 2

• 44 → 44 mod10 = 4

• 68 → 68 mod10 = 8 ∴ collision

∴ new position = 5

• 66 → 66 mod10 = 6

•

	key	chain
0	20	-1
1	11	-1
2	98	5
3	33	-1
4	44	-1
5	68	-1
6	66	-1
7		
8	88	2
9	79	-1

(iii) • 22 → 22 mod 10 = 2 ∴ collision

∴ new position = 2

→ chain changed from 88 → 98 → 68 to 88 → 68 → 98

	key	chain
0	20	-1
1	11	-1
2	22	-1
3	33	-1
4	44	-1
5	68	7
6	66	-1
7	98	-1
8	88	5
9	79	-1