**LC#13:ROMAN TO INTEGER:**

**->push all into the hashmap**

**->take the curr element in the string and compare**

**-> if curr element is smaller than next -> subtract the current value skip the rem loop**

**-> else just add the value**

class Solution {

    public int romanToInt(String s) {

        Map<Character,Integer> hm = new HashMap();

        hm.put('I',1);

        hm.put('V' , 5);

        hm.put('X',10);

        hm.put('L',50);

        hm.put('C',100);

        hm.put('D' , 500);

        hm.put('M' ,1000);

        int res = 0 ,n = s.length();

        for(int i = 0 ; i< n ; i++)

        {

            int curr = hm.get(s.charAt(i));

            if(i+1<n)

            {

                int next = hm.get(s.charAt(i+1));

                if(curr<next)

                {

                    res = res-curr;

                    continue;

                }

            }

            res = res+curr; //outside the i+1<n

        }

        return res;

    }

}

**VICE VERSA:**

**->here all the possible below 1000 need otherwise it would result in 54-> VIIIII**

class Solution {

    public String intToRoman(int num) {

         Map <Integer,String> hm = new LinkedHashMap();

         StringBuilder sb = new StringBuilder();

        hm.put(1000, "M");

        hm.put(900,  "CM");

        hm.put(500,  "D");

        hm.put(400,  "CD");

        hm.put(100,  "C");

        hm.put(90,   "XC");

        hm.put(50,   "L");

        hm.put(40,   "XL");

        hm.put(10,   "X");

        hm.put(9,    "IX");

        hm.put(5,    "V");

        hm.put(4,    "IV");

        hm.put(1,    "I");

        for(Map.Entry<Integer , String> val : hm.entrySet())

        {

            int key = val.getKey();

            String value =  val.getValue();

            while(num>=key)

            {

                num = num-key;

                sb.append(value);

            }

        }

        return sb.toString();

    }

}