```java
//Roman3.java MrG 2018.0320
public class Roman3 extends Object implements Comparable<Roman3>
{
    private int[] digits;
    private final String[][] table={
        {""    ,""    ,""    ,""     ,""    ,""  },
        {"i"   ,"x"   ,"c"   ,"m"    ,"X"   ,"C"},
        {"ii"  ,"xx"  ,"cc"  ,"mm"   ,"XX"  ,"CC"},
        {"iii" ,"xxx" ,"ccc" ,"mmm"  ,"XXX" ,"CCC"},
        {"iv"  ,"xl"  ,"cd"  ,"mV"   ,"XL"  ,"CD"},
        {"v"   ,"l"   ,"d"   ,"V"    ,"L"   ,"D"},
        {"vi"  ,"lx"  ,"dc"  ,"Vm"   ,"LX"  ,"DC"},
        {"vii" ,"lxx" ,"dcc" ,"Vmm"  ,"LXX" ,"DCC"},
        {"viii","lxxx","dccc","Vmmm" ,"LXXX","DCCC"},
        {"ix"  ,"xc"  ,"cm"  ,"mX"   ,"XC"  ,"CM"}
    };
    public Roman3(int num)
    {
        digits = new int[6];
        for(int i=0; i<6; i++)
        {
            digits[i]=num%10;
            num/=10;
        }

    }
    public String getDigits()
    {
        String temp="";
        for(int n:digits)
        {
            temp+=n;
        }
        return temp;
    }
    public int getNum()
    {
        int temp=0;

        for(int i=0; i<6; i++)
        {
            temp+=digits[i]*Math.pow(10,i);
        }

        return temp;
```

```java
46          }
47          public int compareTo(Roman3 other)
48          {
49              return this.getNum()-other.getNum();
50          }
51          public boolean equals(Object other)
52          {
53              Roman3 temp = (Roman3)other;
54              return temp.compareTo(this)==0;
55          }
56          public String toString()
57          {
58              String temp="";
59
60              for(int r=1; r<10; r++)
61              {
62                  for(int c=0; c<6; c++)
63                  {
64                      if(digits[c]==r)
65                      {
66                          temp=table[r][c]+" "+temp;
67                          //temp+=table[r][c];
68                      }
69                  }
70              }
71
72              return temp;
73          }
74          public Roman3 add(Roman3 other)
75          {
76              return new Roman3(getNum()+other.getNum());
77          }
78          public Roman3 sub(Roman3 other)
79          {
80              return new Roman3(getNum()-other.getNum());
81          }
82          public Roman3 mul(Roman3 other)
83          {
84              return new Roman3(getNum()*other.getNum());
85          }
86      }
```