# 3. Bryce

**Program Name: Bryce.java** **Input File: bryce.dat**

Bryce has taken an interest in Roman numerals beyond those appearing on some clock, watch face or sundial. The Olympics and the NFL Super Bowl are probably the most visible users of Roman numerals, but they also occur in books as chapter numbers and preface page numbers, movie titles and years of production, building cornerstones with the year of construction, generational name suffixes like James Bond IV, and more.

Bryce has asked your team to write a program to help him check his practice work. He starts with whole numbers and then tries to write the equivalent Roman numerals. There have been numerous variations throughout the ages, but the modern interpretation is limited to the following symbols and rules.

| Symbol | M | D | C | L | X | V | I |
|---|---|---|---|---|---|---|---|
| Value | 1000 | 500 | 100 | 50 | 10 | 5 | 1 |

The basic rule is that the symbols appear in order of descending values and, in the simplest form, values of symbols are added together so MMXVIII = 1000 + 1000 + 10 + 5 + 1 + 1 + 1 = 2018.

To keep the length of symbols as short as possible, DD is always just M, CCCCC is always just D, LL is always just C, XXXXX is always just L, VV is always just X, and IIIII is always just V. Only M, C, X, and I can be repeated.

The subtraction rule also shortens the length of some symbol combinations and occurs when a symbol of lower value appears before a symbol of higher value. However, they are allowed in limited situations:

- CM = 1000 – 100 = 900 replaces DCCCC for 900 and then CM cannot be followed by any more Cs or a D.
- XC = 100 – 10 = 90 replaces LXXXX for 90 and then XC cannot be followed by any more Xs or an L.
- IX = 10 – 1 = 9 replaces VIIII and then IX cannot be followed by any more Is or a V.
- IV = 5 – 1 = 4 replaces IIII and IV cannot be followed by any more Is.

**Input:** An unknown number of positive integers with values < 5000, each on a separate line.

**Output:** For each input value, display a single line containing the number followed by a colon (:) and then the resulting Roman numeral with no other spacing or extraneous characters.

**Sample input:**

```
2018
1776
1492
1984
2649
9
```

**Sample output:**

```
2018:MMXVIII
1776:MDCCLXXVI
1492:MCCCCXCII
1984:MCMLXXXIV
2649:MMDCXXXXIX
9:IX
```

```
String str = "";
int x = scan.nextInt();
int y = 0;
while(x>0){
  if(x/1000>0){
    y = x/1000;
    while(y>0){
      str = str + "M"
      y--;
    }
    x = x-(x/1000*1000)
  }
  if(x/100>0){
    y = x/100;
    while(y>0);
    str =
```