

```

29
30 romans = {1000: 'M', 900: 'CM', 500: 'D', 400: 'CD',
31           100: 'C', 90: 'XC', 50: 'L', 40: 'XL', 10: 'X', 9: 'IX', 5: 'V', 4: 'IV', 1: 'I'}
32
33
34 roman = [(1000, 'M'), (900, 'CM'), (800, 'DCCC'), (700, 'DCC'), (600, 'DC'), (500, 'D'), (400, 'CD'), (300, 'CCC'),
35          (200, 'CC'), (100, 'C')],
36          [(90, 'XC'), (80, 'LXXX'), (70, 'LXX'), (60, 'LX'), (50, 'L'), (40, 'XL'), (30, 'XXX'), (20, 'XX'), (10, 'X')],
37          [(9, 'IX'), (8, 'VIII'), (7, 'VII'), (6, 'VI'), (5, 'V'), (4, 'IV'), (3, 'III'), (2, 'II'), (1, 'I')]]
38
39
40 def decToRoman(numStr):
41     n = int(numStr)
42     result = ''
43     for value in sorted(romans.keys(), reverse=True):
44         while n >= value:
45             result += romans[value]
46             n -= value
47     return str(result)
48
49
50 def romanToDec(numStr):
51     result = 0
52     for i in range(0, len(roman)):
53         for value, letter in roman[i]:
54             idx = numStr.find(letter)
55             if idx == 0:
56                 result += value
57                 numStr = numStr[len(letter):len(numStr)]
58     return str(result)

```

사전과 리스트 튜플을 만들어서 사전은 reverse=True를 통해 정렬해준 후 while문을 작동시켰고, 리스트 튜플 형식은 바로 for문을 이용해서 roman 숫자를 아라비아 숫자로 변형시켰습니다.

```

functionList = [
    'factorial (!)',
    '-> binary',
    'binary -> dec',
    '-> roman',
]

```

```

21
22 functionList = [
23     'factorial (!)',
24     '-> binary',
25     'binary -> dec',
26     '-> roman',
27     'roman -> dec'
28 ]

```

keypad에는 roman → dec 항목을 추가해줘서 계산기에 나타나도록 했습니다.

```

129         elif key == functionList[4]:
130             n = self.display.text()
131             value = calcFunctions.romanToDec(n)
132             self.display.setText(str(value))

```

마지막으로 mycalc11 파일에 functionlist를 불러올 수 있게 코드를 추가해주었습니다.