

```

112         elif key in constantList:
113             ftion = ['3.141592', '3E+8', '340', '1.5E+8']
114             for calclist in range(0, 4):
115                 if key in constantList[calclist]:
116                     self.display.setText(self.display.text() + ftion[calclist])
117
118         elif key in functionList:
119             n = eval(self.display.text())
120             val = [calcFunctions.factorial(n), calcFunctions.decToBin(n),
121                  calcFunctions.binToDec(n), calcFunctions.decToRoman(n)]
122             for calcftion in range(0, 4):
123                 if key in functionList[calcftion]:
124                     value = val[calcftion]
125                     self.display.setText(str(value))
126
127         else:
128             self.display.setText(self.display.text() + key)
129
130

```

list와 for문을 이용해서 길었던 코드를 반 이상 줄였습니다.

```

76         operator = ['*', '+', '-', '.', '/']

```

괄호를 제외한 operator 리스트를 만들어 주고,

```

98
99         elif self.display.text() == '' and key in operator:
100             pass
101
102         elif self.display.text() != '' and self.display.text()[-1] in operator:
103             if self.display.text()[-1] == '*' and self.display.text()[-2] != '*' and key == '*':
104                 self.display.setText(self.display.text() + key)
105
106             elif self.display.text()[-1] == '/' and self.display.text()[-2] != '/' and key == '/':
107                 self.display.setText(self.display.text() + key)
108
109         elif key not in operator:
110             self.display.setText(self.display.text() + key)

```

수식 연산이 여러 번 입력되어 생기는 오류를 최대한 줄이기 위해서 처리를 해 주었습니다.