```
module
Shebang line: /usr/bin/python3
Encoding: utf-8
    from sys
    argv
    from PyQt5
    QtCore, QtGui, QtWidgets
     ui
    from bigint
    GCD, BigInt
    Класс главной формы
 class
 LongArithmeticCalc(QtWidgets.QMainWindow, ui.Ui_MainWindow)
  def
   _init___(self)
   super().__init__()
   self.setupUi(self)
   self.error_message = 'Ошибка ввода :('
      Events кнопок меню описания и выбора файла
   self.addition.clicked.connect(self.addition_clicked)
   self.subtraction.clicked.connect(self.subtraction_clicked)
   self.multiplication.clicked.connect(self.multiplication_clicked)
   self.division.clicked.connect(self.division_clicked)
   self.exponentiation.clicked.connect(self.exponentiation_clicked)
   self.root.clicked.connect(self.root_clicked)
   self.gdc.clicked.connect(self.gdc_clicked)
                                                                      НОД
   self.remainder.clicked.connect(self.remainder_clicked)
                                                                       0CT
  def
  get_nums(self)
   self.first_num_edit.toPlainText(), self.second_num_edit.toPlainText()
  def
  data_validation(self, a)
                except
   try
                ValueError
     int(a)
                 False
    🗭 True
  addition_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
   self.result.setText(self.error_message)
                                             res = str(BigInt(first_num) + BigInt(second_num))
                                             self.result.setText(res)
  def
  subtraction_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
   self.result.setText(self.error_message)
                                             res = str(BigInt(first_num) - BigInt(second_num))
                                             self.result.setText(res)
  def
  multiplication_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
   self.result.setText(self.error_message)
                                             res = str(BigInt(first_num) * BigInt(second_num))
                                             self.result.setText(res)
  def
  division_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
                                             res = str(BigInt(first_num) / BigInt(second_num))
   self.result.setText(self.error_message)
                                             self.result.setText(res)
  exponentiation_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
   self.result.setText(self.error_message)
                                             res = str(BigInt(first_num).bipow(int(second_num)))
                                             self.result.setText(res)
  def
  root_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
   self.result.setText(self.error_message)
                                             res = str(BigInt(first_num).birt(int(second_num)))
                                             self.result.setText(res)
                                                                                                 НОД
  gdc_clicked(self)
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
                                             res = str(GCD(BigInt(first_num), BigInt(second_num)))
   self.result.setText(self.error_message)
                                             self.result.setText(res)
  def
  remainder_clicked(self)
                                                                                             OCT
   first_num, second_num = self.get_nums()
    self.data_validation(first_num) and self.data_validation(second_num)
   self.result.setText(self.error_message)
                                             res = str(BigInt(first_num) % BigInt(second_num))
                                             self.result.setText(res)
 def
main()
  app = QtWidgets.QApplication(argv)
 main_window = LongArithmeticCalc()
  main_window.show()
  app.exec_()
   _name___ == "___main__
                  main()
```