```ruby
#!/usr/bin/ruby
#
# THOU SHALT EXECUTE MINE COMMANDMENTS HEREIN ACCORDING TO MY LAWS, WHICH ARE DEAR TO ME.
# THUS: ./KJV.rb KJV.txt fire flame -d lord god
# AND WE SHALL LOOK UPON THE FRUITS OF THIS THY LABOR AND WE SHALL JUDGE THEE THUS.
#



class KJV

  # IN THE BEGINNING WAS THE INIT. AND THE INIT RECEIVES A HOLY LOGOS
  # CONTAINING THE COMMANDMENTS BEGOTTEN OF HE WHO COMMANDS.
  def initialize( logos = [ ] )
    if logos.empty?
      return nil
    else
      @file, @terms, @dcs, @book, @verno, @verse = nil
      self.burning_bush(logos)
      if @file
        logos.delete(@file)
        self.deuterocanon(logos.join(" "))
        if !@dcs.nil?
          @dcs.each { |dc| logos.delete(dc) }
          logos.delete("-d")
        end
        @terms = logos
      end
    end
  end
  attr_reader :book, :verno, :verse
  attr_writer :book, :verno, :verse



  # AND HE WHO COMMANDS CALLED TO HIS MESSENGER. AND HE SPOKE HIS LOGOS
  # AND HIS MESSENGER WENT TO THE WILDERNESS TO LOCATE THE BOOK OF THE LAW.
  def burning_bush( logos = [ ] )
    return nil if logos.empty?
    file = nil
    logos.each { |book| file = book if File.exists?(book) }
    @file = file if file
  end



  # AND HE WHO COMMANDS MAYBE BUILD CONTINGENCIES IN HIS LAW. FOR HIS LAW
  # IS OUR LAW AND OUR LAW IS HIS WORD AND HIS WORD EMBODIES HIS DESIRE.
  def deuterocanon(logos = nil)
    return nil if logos.nil?
    dcs = nil
    if (dc = logos.match(/\-d (.*)$/))
      dcs = dc[1].split(" ")
    end
    @dcs = dcs if dcs
  end



  # AND HIS SOLDIERS AND HIS SCRIBES AND HIS ENUMERATORS WERE FAITHFUL
  # TO HIS LAW. AND HIS PEOPLE READ THE BOOK OF HIS LAW LINE BY LINE.
  # AND THEY MADE MUCH MEANING OF THE LINES OF THE BOOK OF HIS LAW.
  def genesis
    return nil if (@file.nil? or @terms.nil?)

    @book, @verno, @verse, bkchk = nil, nil, '', 0
    rls = File.open(@file, "r")
    rls.each do |rl|
      rl = rl.rstrip

      if (rl == '' or rl.empty?)
        bkchk += 1
      else

        if (bkchk >= 3 and self.know_thy_line_and_know_thy_book(rl))
          self.prophecy
          self.anoint(rl)
        else
        chk = self.line_for_line(rl)
        if (chk == rl)
          @verse << " " + chk
        else
          @verse = chk
        end
      end
      bkchk = 0

    end

  end
end



# AND HIS PEOPLE GATHERED THE LINES OF HIS LAW AND PUT THEM IN VERSES.
def line_for_line(line = nil)
  return nil if line.nil?

  chk = self.order_by_ordering(line)
  ret = (chk.nil?) ? line : self.line_for_line(chk)

  return ret
end



# AND EACH VERSE OF HIS LAW WAS READ AND NUMBERED.
# AND EACH VERSE OF HIS LAW WAS NUMBERED AND ACCOUNTED FOR.
# FOR HIS PEOPLE ARE A PEOPLE STIFF-NECKED AND FAITHFUL TO HIS LAW.
def order_by_ordering(line = nil)
  return nil if line.nil?

  ret = nil
  if (m = line.match(/^(.*?)([0-9]+:[0-9]+)(.*)$/))
    @verse << " " + m[1]
    self.prophecy
    @verno = m[2]
    ret = m[3]
  end

  return ret
end



# AND HIS PEOPLE SEPARATED THE CHAFF FROM THE GRAIN. FOR THE GRAIN
# IS GOOD IN HIS SIGHT BUT THE CHAFF CHAFES SORE HIS COMMANDMENTS.
def chaff_from_grain
  ret = nil
  @terms.each { |term| ret = true if @verse.include?(term) }
  return ret
end



# AND HIS LAWS HE PUT IN BOOKS. AND HIS PEOPLE SOUGHT HIS BOOKS AND
# SOUGHT TO KNOW THEIR NAMES. AND THOSE NAMES THEY PROCLAIMED BOLDLY.
def know_thy_line_and_know_thy_book(line = nil)
  return nil if line.nil?
  ret = (((line.match(/\AThe [-A-Z0-9a-z: ]+\Z/)) or
         (line.match(/\A[A-Z]+[a-z]+\Z/))) and
         (!self.order_by_ordering(line))) ? true : nil
  return ret
end



# AND HE CREATED BOOKS AND BOOKS OF BOOKS. AND THE NAMES OF THE
# GREATER BOOKS SHALL BE MADE GREATER IN THE EYES OF HIS PEOPLE.
def anoint(title = nil)
  return nil if title.nil?
  @book = title
  self.the_greater_over_the_many if @book.include? 'Testament'
end
```

```
  # AND THE NAMES OF HIS BOOKS SHALL BE WRAPPED IN A WRAP OF BOLD.
  # AND THE GREATER NAMES SHALL BE WRAPPED IN A GREATER WRAP OF BOLD.
  def the_greater_over_the_many
    if @book.include? "Testament"
      puts "<h1 var=\"testament\">#{@book}</h1>"
    else
      puts "<h1>#{@book}</h1>"
    end
    @book = nil
  end



  # AND HIS PROPHETS SPAKE IN THE WILDERNESS FOR ONLY THOSE LAWS
  # BEHOLDEN TO THE WORDS OF HIS LAW WERE TAKEN TO HIS PEOPLE.
  def prophecy
    self.proclaim if self.chaff_from_grain
  end



  # AND HIS PROPHETS PROCLAIMED HIS LAW TO HIS PEOPLE.
  # AND HIS SOLDIERS AND SCRIBES, EVEN ALL HIS PEOPLE
  # FAITHFUL AND STIFF-NECKED, RECEIVED THE WORDS OF HIS LAW.
  def proclaim
    out = ''

    self.the_greater_over_the_many if !@book.nil?

    @verse.split(" ").each do |word|
      @terms.each do |term|
        word = self.linens_for_laws(word, term) if word.include? term
      end
      if @dcs
        @dcs.each do |term|
          word = self.linens_for_laws(word, term) if word.include? term
        end
      end
      out << word + " "
    end

    puts "<p><span class=\"color\" var=\"vn\">#{@verno}</span> " + out.gsub("\n", " ").squeeze(" ").rstrip + "</p>"
    @verse = ''
  end



  # FOR HIS WORDS WERE WRAPPED IN A WRAP OF MANY COLORS.
  # FOR HIS PEOPLE LOVED THINGS COLORED AND SHINY.
  def linens_for_laws(word = nil, term = nil)
    return nil if (word.nil? or term.nil?)

    t = term.downcase
    if (m = word.match(/([^A-Za-z]*)([A-Za-z]+)([^A-Za-z]*.*)/))
      ww = "#{m[1]}<span class=\"color\" var=\"#{t}\">#{m[2]}</span>#{m[3]}"
    else
      ww = "<span class=\"color\" var=\"#{t}\">#{word}</span>"
    end

    return ww
  end

end



# LET IT BE.
KJV.new(ARGV).genesis
```