```
#
class MaidenHead
  attr_accessor :mhl, :long, :lat

  def i_vars(locator)
    @mhl = locator.upcase.bytes
    @long = -180.0
    @lat = -90.0
  end

  def locator4_to_lat
    (@mhl[1] - 65) * 10.0 +
      @mhl[3] - 48
  end

  def locator4_lat_total
    @lat = locator4_to_lat + 0.5 - 90
  end

  def locator6_to_lat
    @mhl[5] - 65
  end

  def locator6_lat_total
    @lat = locator4_to_lat +
           (locator6_to_lat + 0.5) / 24.0 - 90
  end

  def locator8_to_lat
    @mhl[7] - 48
  end

  def locator8_lat_total
    @lat = locator4_to_lat +
           locator6_to_lat / 24.0 +
           (locator8_to_lat + 0.5) / 240.0 - 90
  end

  def locator10_to_lat
    @mhl[9] - 65
  end

  def locator10_lat_total
    @lat =
    locator4_to_lat +
    locator6_to_lat / 24.0 +
    locator8_to_lat / 240.0 +
    (locator10_to_lat + 0.5) / 240.0 / 24.0 - 90
end

def lat_total
  locator4_lat_total if @mhl.size == 4
  locator6_lat_total if @mhl.size == 6
  locator8_lat_total if @mhl.size == 8
  locator10_lat_total if @mhl.size == 10
end

def locator_to_lat
  lat_total
  Exception.new('Invalid locator format')
end

def locator2_to_long
  @mhl[0] - 65
end

def locator4_to_long
  @mhl[2] - 48
end

def locator4_long_total
  @long =
    locator2_to_lat * 20 +
    (locator4_to_long + 0.5) * 2.0 - 180
end

def locator6_to_long
  @mhl[4] - 65
end

def locator6_long_total
  @long =
    locator2_to_long * 20 +
    locator4_to_long * 2.0 +
    (locator6_to_long + 0.5) / 12.0 - 180
end

def locator8_to_long
  @mhl[6] - 48
  end

  def locator8_long_total
    @long =
      locator2_to_long * 20.0 +
      locator4_to_long * 2.0 +
      locator6_to_long / 12.0 +
      (locator8_to_long + 0.5) / 120.0 - 180
  end

  def locator10_to_long
    @mhl[8] - 65
  end

  def locator10_long_total
    @long =
      locator2_to_long * 20.0 +
      locator4_to_long * 2.0 +
      locator6_to_long / 12.0 +
      locator8_to_long / 120.0 +
      (locator10_to_long + 0.5) / 2880 - 180
  end

  def long_total
    locator4_long_total if @mhl.size == 4
    locator6_long_total if @mhl.size == 6
    locator8_long_total if @mhl.size == 8
    locator10_long_total if @mhl.size == 10
  end

  def locator_to_lng
    long_total
    Exception.new('Invalid locator format')
  end

  def locator_to_lat_lng(locator)
    i_vars(locator)
    locator_to_lng
    locator_to_lat
  end
end

mh = MaidenHead.new
locator = 'EN51pw07UJ'
```

```
mh.locator_to_lat_lng(locator)
p mh.mhl
p mh.long
p mh.lat

# from http://unclassified.software/en/source/maidenheadlocator

# Long = (locator[0] - 'A') * 20 +
#        (locator[2] - '0' + 0.5) * 2 - 180

# Long = (locator[0] - 'A') * 20 +
#        (locator[2] - '0') * 2 +
#        (locator[4] - 'A' + 0.5) / 12 - 180

# Long = (locator[0] - 'A') * 20.0 +
#        (locator[2] - '0') * 2.0 +
#        (locator[4] - 'A') / 12.0 +
#        (locator[6] - '0' + 0.5) / 120.0 - 180

# Long = (locator[0] - 'A') * 20.0 +
#        (locator[2] - '0') * 2.0 +
#        (locator[4] - 'A') / 12.0 +
#        (locator[6] - '0') / 120.0 +
#        (locator[8] - 'A' + 0.5) / 120.0 / 24.0 - 180

# Lat = (locator[1] - 'A') * 10 +
#       (locator[3] - '0' + 0.5) - 90

# Lat = (locator[1] - 'A') * 10 +
#       (locator[3] - '0') +
#       (locator[5] - 'A' + 0.5) / 24 - 90

# Lat = (locator[1] - 'A') * 10.0 +
#       (locator[3] - '0') +
#       (locator[5] - 'A') / 24.0 +
#       (locator[7] - '0' + 0.5) / 240.0 - 90

# Lat = (locator[1] - 'A') * 10.0 +
#       (locator[3] - '0') +
#       (locator[5] - 'A') / 24.0 +
#       (locator[7] - '0') / 240.0 +
#       (locator[9] - 'A' + 0.5) / 240.0 / 24.0 - 90
```