

```

#-----
#
# 双曲線起動（外向き：脱出系） 計算
#
#           Programmer:H.Endo
#       Create Date:2021/03/09
#
#-----

import math

#
# 定数変数 クラス
#

class ConstSet:
    c=299792458/1000*3600 #光速(km/h)
    U=7.42426e-31      #宇宙エネルギー一定数(km/kg)

    M = 1.9891e30      #太陽質量(kg)
    m = 5.97219e24     #地球質量(kg)
    mp = 1000          #宇宙船質量(kg)
    Re = 149598262     #太陽-地球間距離
    r = 6378.137       #地球半径

    #近点
    rp = 6378.137 + 260.9 #脱出（近点）(km)
     $\theta = 0.47076$  #脱出角度(rad)  $\theta = 26.97242^\circ$ 
    vX = 44257.98535     #脱出速度(km/h)
    vHr=0 # 基準軌道方向速度
    vVr=0 # 中心天体方向速度
    AHr=0 # 基準軌道方向動的エネルギー
    AVr=0 # 中心天体逆方向動的エネルギー
    Sa=0 #

    #半交軸
    a = 0 #脱出（近点）(km)
     $\theta a = 0$  #脱出角度(rad)  $\theta = 26.97242^\circ$ 
    va = 44257.98535     #脱出速度(km/h)
    vHa=0 # 基準軌道方向速度
    vVa=0 # 中心天体方向速度

```

```
AHa=0 # 基準軌道方向動的エネルギー  
AVa=0 # 中心天体逆方向動的エネルギー  
Aa=0
```

```
#影響圏
```

```
R = 0 #脱出 (近点) (km)  
θ R = 0 #脱出角度(rad) θ=26.97242°  
vR = 0 #脱出速度(km/h)  
vHR=0 # 基準軌道方向速度  
vVR=0 # 中心天体方向速度  
AHR=0 # 基準軌道方向動的エネルギー  
AVR=0 # 中心天体逆方向動的エネルギー  
AR=0
```

```
#
```

```
# 定数
```

```
#
```

```
def sayU(self):
```

```
    print("光速 c = {0} km/h , 宇宙エネルギー定数 U = {1} km/kg".format(self.c,self.U))
```

```
#
```

```
# 変数
```

```
#
```

```
def sayM(self):
```

```
    print("")
```

```
    print("太陽質量          M = %.5e kg"%(self.M))
```

```
    print("地球質量          m = %.5e kg"%(self.m))
```

```
    print("宇宙船質量        mp = %.5e kg"%(self.mp))
```

```
    print("太陽-地球間距離    Re = %.5e km"%(self.Re))
```

```
    print("地球半径          r = %.5e km"%(self.r))
```

```
    print("")
```

```
    print("<脱出(近点) r>")
```

```
    print("距離          rp = %.5e km"%(self.rp))
```

```
    print("速度          vX = %.5e km/h"%(self.vX/3600))
```

```
    print("角度          θ = %.5e 度"%(self.θ*180/math.pi))
```

```
    print("速度 (基準軌道) vHr = %.5e km/h"%(self.vHr/3600))
```

```
    print("速度 (中心天体) vVr = %.5e km/h"%(self.vVr/3600))
```

```
    print("エネ (基準軌道) AHr = %.5e je"%(self.AHr))
```

```
    print("エネ (中心天体) AVr = %.5e je"%(self.AVr))
```

```
    print("")
```

```

print("<脱出(半交軸) a>")
print("距離          a  = %.5e km"%(self.a))
print("速度          va = %.5e km/h"%(self.va/3600))
print("角度          θ a = %.5e 度"%(self.θ a*180/math.pi))
print("エネ (基準軌道) AHa = %.5e je"%(self.AHa))
print("エネ (中心天体) AVa = %.5e je"%(self.AVa))
print("速度 (基準軌道) vHa = %.5e km/h"%(self.vHa/3600))
print("速度 (中心天体) vVa = %.5e km/h"%(self.vVa/3600))

```

```
print("")
```

```

print("<脱出(影響圏半径) R>")
print("距離          R  = %.5e km"%(self.R))
print("速度          vR = %.5e km/h"%(self.vR/3600))
print("角度          θ R = %.5e 度"%(self.θ R*180/math.pi))
print("エネ (基準軌道) AHR = %.5e je"%(self.AHR))
print("エネ (中心天体) AVR = %.5e je/h"%(self.AVR))
print("速度 (基準軌道) vHR = %.5e km/h"%(self.vHR/3600))
print("速度 (中心天体) vVR = %.5e km/h"%(self.vVR/3600))

```

```
print("")
```

```

#
# 速度 クラス
#

```

```
class Velocity:
```

```
    cs=ConstSet()
```

```
    #
```

```
    # 地球公転速度
```

```
    #
```

```
    def VxSep(self):
```

```
        μ = self.μ (cs.m,cs.mp)          # 重力定数
```

```
        #近点
```

```
        cs.vHr = cs.vX * math.cos(cs.θ)   # 基準軌道方向速度
```

```
        cs.vVr = cs.vX * math.sin(cs.θ)   # 中心天体逆方向速度
```

3/5ページ

```

cs.AHr = self.A(cs.mp,cs.vHr)    # 基準軌道方向動的エネルギー
cs.AVr = self.A(cs.mp,cs.vVr)    # 中心天体逆方向動的エネルギー
cs.Sa = cs.AVr

#半交軸
cs.AVa = cs.Sa                    # 半交軸(a)の静的エネルギー
cs.AHa = 2*cs.AVa                # 基準軌道方向動的エネルギー
cs.vVa = self.v(cs.AVa,cs.mp)
cs.vHa = self.v(cs.AHa,cs.mp)
cs.a =  $\mu$  / cs.AVa             # 半交軸距離(a)
cs.Aa = 2*cs.Sa + cs.AVr        # 半交軸(a)の動的エネルギー
cs.va = self.v(cs.Aa,cs.mp)     # 半交軸(a)の動的エネルギー
cs. $\theta$ a = math.atan(cs.vVa/cs.vHa)

#影響圏
cs.AVR = cs.Sa                   # 中心天体逆方向動的エネルギー
cs.AHR = cs.Sa                   # 基準軌道方向動的エネルギー
cs.vVR = self.v(cs.AVR,cs.mp)
cs.vHR = self.v(cs.AHR,cs.mp)
cs.SR = cs.Sa/2
cs.R =  $\mu$  / (cs.SR)              # 影響圏距離(a)
cs.AR = 2*cs.SR + cs.AVa        # 影響圏(a)の動的エネルギー
cs.vR = self.v(cs.AR,cs.mp)     # 影響圏(a)の動的エネルギー
cs. $\theta$ R = math.atan(cs.vVR/cs.vHR)

#
# 速度->動的エネルギー
#
def A(self,m,v):
    return m*v**2                # A=mv^2

#
# 運動エネルギー->速度
#
def v(self,A,m):
    return (A/m)**0.5            # v=sqrt(A/m)

#
# 静的エネルギー
#
def S(self,M,m,r,n):
    Em = m * cs.c**2             # Em = mp x c^2

```

```
ac = cs.U * (M + m)      # ac = U x (m + mp)
Sr = n * Em * ac / r     # Sr = Em x ac / r
return Sr
```

```
#
# 重力定数
#
def  $\mu$ (self,M,m):
    Em = m * cs.c**2      # Em = mp x c^2
    ac = cs.U * (M + m)   # ac = U x (m + mp)
     $\mu$  = Em * ac          #  $\mu$  = Em x ac
    return  $\mu$ 
```

```
#
# Main Routine
#
```

```
cs = ConstSet()          #定数変数クラス
vl = Velocity()          #速度クラス
```

```
cs.sayU()                #手数印刷
vl.VxSep()
cs.sayM()
```