

Question 1

Convert the RA from hours-minutes-seconds to decimal hours and the Dec from degrees-minutes-seconds to decimal degrees

```
In [19]: Ra_hrs = 5
RA_mins = 55
Ra_s = 10

Ra_decimal = (Ra_hrs) + (RA_mins / 60) + (Ra_s / 3600) # RA-> decimal hours

# print(Ra_decimal)

Dec_degree = 7
Dec_mins = 24
Dec_sec = 26

Dec_degrees = Dec_degree + (Dec_mins/60) + (Dec_sec/3600)

print("Value of alpha ->" ,Ra_decimal,"\nValue of delta ->", Dec_degrees)
```

Value of alpha -> 5.919444444444445

Value of delta -> 7.407222222222223

Question 2

Compute the local sidereal time (LST) of rise and set for our latitude

```
In [12]: import math
import numpy as np

latitude = 40.112

latitude_r = math.radians(latitude)
dec_rad = math.radians(Dec_degrees)

cos_h = -(math.sin(latitude_r) * math.sin(dec_rad))/(math.cos(latitude_r) * math.cos(dec_rad))

print ("cos_h ->" ,cos_h)

H_rad = math.acos(cos_h)

H_degrees = math.degrees(H_rad)

H_hrs = H_degrees/15

LST_rise = Ra_decimal - H_hrs
LST_set = Ra_decimal + H_hrs

print("LST rise value ->", LST_rise , " \nLST Set value ", LST_set )
```

```
cos_h -> -0.1095213433109864
LST rise value -> -0.4997371457591937
LST Set value 12.338626034648083
```

Question 3

Add or subtract 24 from each number as needed to get values between 0 and 24

```
In [20]: LST_rise_24 = LST_rise + 24 if LST_rise < 0 else LST_rise
LST_set_24 = LST_set if LST_set < 0 else LST_set

print("LST rise value adjusted ->", LST_rise_24 , " \nLST Set value adjusted ", LST_set_24)

LST rise value adjusted -> 23.500262854240805
LST Set value adjusted 12.338626034648083
```

Question 4 Convert these times to Greenwich sidereal time for our longitude

Question 5 Again add or subtract 24 as needed.

```
In [21]: GST_time_rise = LST_rise_24 - (-88.221/15)
GST_time_set = LST_set_24 - (-88.221/15)

GST_time_rise_adj = GST_time_rise + 24 if GST_time_rise < 0 else GST_time_rise
GST_time_rise_adj = GST_time_rise_adj - 24 # since it is greater than 24
GST_time_set_adj = GST_time_set + 24 if GST_time_set < 0 else GST_time_set

GST_time_rise_adj, GST_time_set_adj

print("GST rise value adjusted ->", GST_time_rise_adj , " \nGST Set value adjusted ", GST_time_set_adj)

GST rise value adjusted -> 5.381662854240805
GST Set value adjusted 18.220026034648082
```

Question 6

Now calculate the Julian date corresponding to the Greenwich date (d,m,y) 30 Jan 2024

Question 7

Convert the Greenwich sidereal time to Universal Time. Compute T and T0 as follows. Add or subtract to get values between 0 and 24. Then compute UT as in the last line:

Question 8

Again add or subtract (multiples of) 24 as needed for UT.

```
In [47]: y = 2024
m = 1
d = 30

A = math.trunc((y-1) / 100)
B = 2 - A + math.trunc(A/4)
C = math.trunc(365.25 * (y - 1))
D = math.trunc(30.6001 * (m + 13))
```

```

JD = B + C + D + d + 1720994.5

print("Value of JD ->", JD)

T = (JD - 2451545.0) / 36525.0

print("T ->", T)

To = 6.697374558 + 2400.051336 * T + 0.000025862 * (T ** 2)

print(To, " <- To")
while (To > 24) :
    To = To - 24
print(To, " <-new To -> reduced to under 24")
UT_rise = 0.9972695663 * (GST_time_rise_adj - To)
UT_set = 0.9972695663 * (GST_time_set_adj - To)

# UT_rise, UT_set, To
# UT_rise_adj = UT_rise
# UT_set_adj = UT_set
# while (UT_rise_adj < 0 ):
#     UT_rise_adj += 24
# while (UT_set_adj < 0) :
#     UT_set_adj += 24

# UT_rise_adj, UT_set_adj
print("Value of UTr :", UT_rise_adj, "\nValue of UTs :", UT_set_adj)

```

```

Value of JD -> 2460339.5
T -> 0.24078028747433264
584.5824266925894 <- To
8.582426692589365 <-new To -> reduced to under 24
Value of UTr : 20.807975635101407
Value of UTs : 9.611284516028062

```

Question 9

Finally, convert to local standard time here in Urbana, then back to h:m:s:

```

In [48]: CST_rise = UT_rise_adj - 6
CST_set = UT_set_adj - 6

def dec_to_hms(CST_time) :
    hours = int(CST_time)
    mins = int((CST_time - hours) * 60)
    sec = math.trunc(((CST_time - hours) * 60 - mins) * 60)
    return hours, mins, sec

CST_rise_hms = dec_to_hms(CST_rise)
CST_set_hms = dec_to_hms(CST_set)

```

```
print("CST rise at ", CST_rise_hms , " \nCST set at :", CST_set_hms)
```

CST rise at (14, 48, 28)

CST set at : (3, 36, 40)