Zubin Mishra
UID: 604644805
CEE/MAE M20
April 18, 2017

Homework 2

1 The Lunar Phase Calculator

1.1 Introduction

The goal in this problem is to develop a program that will calculate the lunar phase for a user-input date. This is accomplished by converting the input date into its Julian Day Number, calculating the elapsed number of days from a known new moon to be calculated. This result is then plugged into an equation to calculate the percent illumination of the moon.

1.2 Model and Methods

The script first establishes some constants and default values (days in the lunar cycle, offset value a = 0). The script then gets the user-input date as strings using the input function:

```
month = input('Please enter the month as MMM (e.g. JAN): ', 's');
month = upper(month);
day = input ('Please enter the day as DD (e.g. 01): ', 's');
year = input('Please enter the year as YYYY (e.g. 2000): ', 's');
```

The script then checks for valid input by checking string length, then checking if str2num results in a number. This is shown for the year input below:

```
if (length(year) ~= 4)
    error('Year input error');
end
yr = str2num(year);
if (length(yr) ~= 1)
    error('Year input error');
end
```

The year value is also made sure to be in range.
The script then uses a switch statement to check that the month input is correct, and checks the day input in similar fashion to the year input, with additional constraints on value depending on month and year (leap years). A month value and the offset value a are also assigned in the switch statement. The beginning of the switch statement is shown below:

```
switch (month)
    case 'JAN'
        if (d > 31)
            error('Day input error')
        end
        a = 1;
        mon = 1;
    case 'FEB'
```

```
        if ((is_leapyear && d > 29) || (~is_leapyear && d > 28))
            error('Day input error')
        end
        a = 1;
        mon = 2;
    case 'MAR'
        if (d > 31)
            error('Day input error')
        end
        mon = 3;
```

The elapsed days from January 1, 1900 are then calculated by converting the input date to its Julian Day Number:

```
y = yr - a + 4800;
m = mon + 12*a - 3;
J = d + floor((153*m + 2)/5) + 365*y + floor(y/4) - floor(y/100)...
    + floor(y/400) - 32045;
d_J = J - J_0;
```

From this, the percent illumination is calculated and a Boolean value is determined to keep track of waxing vs. waning.

```
L = 100*(sin(pi*(mod(d_J, T)/T)))^2;
waxing = (mod(d_J, T)/T) < 0.5;
```

The results are then printed out using fprintf.

1.3 Results and Calculations

When the program is executed, and the user inputs MAY 10 2017, the following output is printed to the screen:

```
MAY 10 2017:
Illumination = 99.5 percent
Waxing
```

When the program is executed, and the user inputs MAY 11 2017, the following output is printed to the screen:

```
MAY 11 2017:
Illumination = 99.9 percent
Waning
```

This is additionally the next date at which this method says the phase switches from ‘waxing’ to ‘waning,’ signifying that the date of the next full moon will be May 10, 2017.

1.4 Discussion

As stated above, this model predicts the date of the next full moon to be May 10, 2017. This could be off by a few hours however. This model assumes a perfectly circular orbit and does not take into account the time of day that the new moon occurred, so it may be off by several hours. This method could be extended and improved in the future by taking account of lunar apogee and perigee.

2 Binary Addition

2.1 Introduction

The goal in this problem is to add two numbers using binary addition. This is accomplished by taking two numbers as user input, converting them from decimal to binary, and then adding them using XOR, AND, and OR logic. The results are displayed in binary and decimal.

2.2 Model and Methods

The script first obtains the decimal input numbers A and B using the input function. A and B are then checked for validity (integer from 0 to 7). This is shown for A below:

```
if (mod(A,1) ~= 0 || A < 0 || A > 7)
    error('A input error');
end
```

A and B are then converted to binary by using arrays of size 3 and a while loop. The way this was done is shown below for A:

```
A_bin = zeros(1, 3);

A_=A;
i = 1;
while (A_ ~= 0)
    A_bin(i) = mod(A_, 2);
    A_ = floor(A_/2);
    i = i + 1;
end
```

This results in the 1's place to be at index 1, 2's place to be at index 2, and 4's to be at index 3. The script then adds the two binary numbers found. This is accomplished by creating two arrays of length 4, one for the sum digits and one for the carryover digits. The script then runs a for loop that goes through each of the 3 digits for the two input numbers and determined sum and carryover digits in the following manner:

```
for i = 1:3
    sum(i) = xor(xor(A_bin(i), B_bin(i)),carryover(i));
    carryover(i+1) = and(A_bin(i), B_bin(i)) || carryover(i);
```

```
end
sum(4) = carryover(4);
```

The last sum digit is determined solely by the carryover digit, so it's assignment is outside the for loop. The sum is converted back to decimal with the following line:

```
dec = sum(1) + sum(2)*2 + sum(3)*4 + sum(4)*8;
```

The results are then output using fprintf.

2.3 Results and Calculations

When the program is executed, and the user inputs 5 and 2, the following output is printed to the screen:

```
The decimals provided are A = 5 and B = 2
Conversion of A to binary: 101
Conversion of B to binary: 010
A plus B in binary is: 0111
A plus B in decimal is: 7
```

When the program is executed, and the user inputs 7 and 7, the following output is printed to the screen:

```
The decimals provided are A = 7 and B = 7
Conversion of A to binary: 111
Conversion of B to binary: 111
A plus B in binary is: 1110
A plus B in decimal is: 14
```

2.4 Discussion

This method could be extended to add any two positive integers by having an array of dynamic size for the binary equivalents of the two input numbers. The lengths of other arrays and for loops could be determined from the length of the longer array, and leading zeros would be added to the shorter array to avoid indexing errors. Converting to binary would also be performed through a loop, with the power being determined by index.