```
#Evan DePosit
#New Beginnings
#capstone
# this file contains functions to convert time strings into integer minutes and back
again

def min_to_time(minTime):

    minutes=minTime%60
    hour=int(minTime/60)
    #print('hour= ', hour)
    #print('minutes= ', minutes)

    if hour == 12:
        timeOfDay='PM'
    elif hour > 12:
        timeOfDay='PM'
        hour= hour-12
    else:
        timeOfDay='AM'

    if minutes==0:
        time=str(hour) + ':' + str(minutes) + '0' + ' ' + timeOfDay
    else:
        time=str(hour) + ':' + str(minutes) + ' ' + timeOfDay

    #print('time= ', time)
    return time

def time_to_min(time):
    #time_to_min('11:15')
    #input time as string
    #output time as minutes
    #test cases
    minutes=999

    fields= time.split(':')
    #print()
    #print(fields)

    #get hour first
    hourStr=fields.pop(0)
    hour=int(hourStr)

    #start by getting time of day from last string in list iff there at all
    if (('am' in fields[-1]) or ('AM' in fields[-1])):
        #print('its morning')
        timeOfDay= 0
    elif(('pm' in fields[-1]) or ('PM' in fields[-1]) and (hour !=12)):
        #print("it's the afternoon")
        #time of day equals minutes that happened in morning that need to be added
        timeOfDay=720
    else:
        #print('no time of day info in string')
        timeOfDay=None

    minStr=fields.pop(0)

    #print('hour string= ', hourStr)
    #print('minutes string= ', minStr)
    #if timeOfDay is not None:
        #print('time of day: ', timeOfDay)
```

```
    #get numbers from sting
    minutes= int(minStr[:2:])

    #print('hour int=', hour)
    #print('minutes int= ', minutes)
    #print('remaining minutes', minStr)
    if timeOfDay is None:
        if(hour >=1 and hour< 7):
            minutes= ((hour +12)*60)+ minutes
        else:
            minutes= (hour*60)+minutes
    else:
        minutes =(60 * hour) + minutes + timeOfDay

    #print('total minutes', minutes)
    return minutes
```