Chapter16_Exercises

June 6, 2017

1 Chapter 16 Exercises

```
In [2]: class Time:
            """Represents the time of day.
            attributes: hour, minute, second
            """

        def time_to_int(time):
            minutes = time.hour * 60 + time.minute
            seconds = minutes * 60 + time.second
            return seconds

        def int_to_time(seconds):
            time = Time()
            minutes, time.second = divmod(seconds, 60)
            time.hour, time.minute = divmod(minutes, 60)
            return time

        def valid_time(time):
            if time.hour < 0 or time.minute < 0 or time.second < 0:
                return False
            if time.minute >= 60 or time.second >= 60:
                return False
            return True

        def add_time(t1, t2):
            assert valid_time(t1) and valid_time(t2)
            seconds = time_to_int(t1) + time_to_int(t2)
            return int_to_time(seconds)

        time = Time()
        time.hour = 11
        time.minute = 59
        time.second = 30
```

1.1 Exercise 16.1

Write a function called mul_time that takes a Time object and a number and returns a new Time object that contains the product of the original Time and the number. Then use mul_time to write a function that takes a Time object that represents the finishing time in a race, and a number that represents the distance, and returns a Time object that represents the average pace (time per mile).

```
In [3]: def mul_time(timey, number):
            time = Time()
            assert valid_time(time)
            seconds = time_to_int(time) * number

            return int_to_time(seconds)

        def race(timey, distance):
            time = Time()
            return mul_time(time, 1/distance)
```

1.2 Exercise 16.2

The datetime module provides time objects that are similar to the Time objects in this chapter, but they provide a rich set of methods and operators. Read the documentation at http: // docs. python. org/ 3/ library/ datetime. html . 1. Use the datetime module to write a program that gets the current date and prints the day of the week. 2. Write a program that takes a birthday as input and prints the user's age and the number of days, hours, minutes and seconds until their next birthday. 3. For two people born on different days, there is a day when one is twice as old as the other. That's their Double Day. Write a program that takes two birthdays and computes their Double Day. 4. For a little more challenge, write the more general version that computes the day when one person is n times older than the other.

```
In [35]: import datetime
         import calendar

         def print_day():
             print(calendar.day_name[datetime.datetime.today().weekday()])

         def age(birthday):
             """birthday is a datetime date object"""
             today = datetime.datetime.today()
             next_birthday = datetime.datetime(today.year, birthday.month, birthday
             if next_birthday < today:
                 next_birthday = datetime.datetime(today.year + 1, birthday.month,

             print((today - birthday).days / 365)
             print(next_birthday - today)

         def double_day(bday1, bday2):
             assert bday1 > bday2
             delta = bday1 - bday2
```

```python
    double = bday1 + delta
    return double

def n_day(bday1, bday2, n):
    assert bday1 > bday2
    delta = bday1 - bday2
    nday = bday1 + delta*(n-1)
    return nday

print_day()
birthday = datetime.datetime(1984,1,1)
age(birthday)
```

```
Tuesday
33.45205479452055
208 days, 7:00:22.482730
```