**Lab 3 - Conditions and Tuples**

**(Part II)**

**Note:** This part of Lab 3 is for you to complete using Notepad++ and Anaconda Prompt.

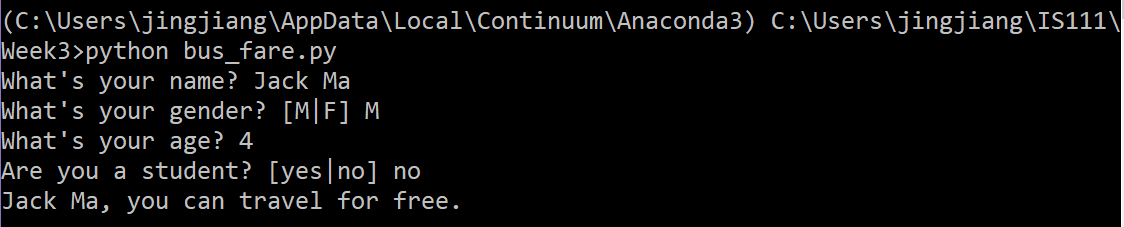
**Q5 [ \*\* ]: Bus Fares**

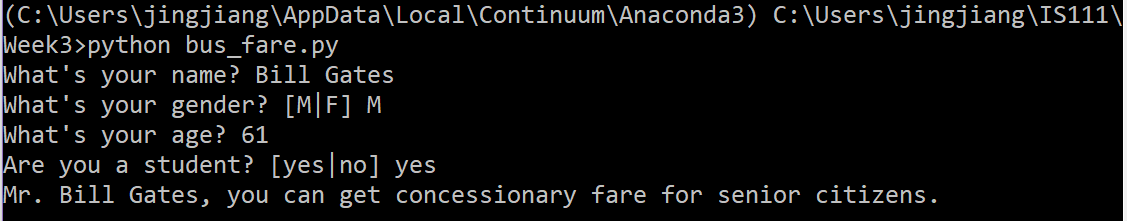
We have provided you with a script called bus\_fare.py. Inside the file we have implemented a function called get\_user\_info. Do not modify this function.

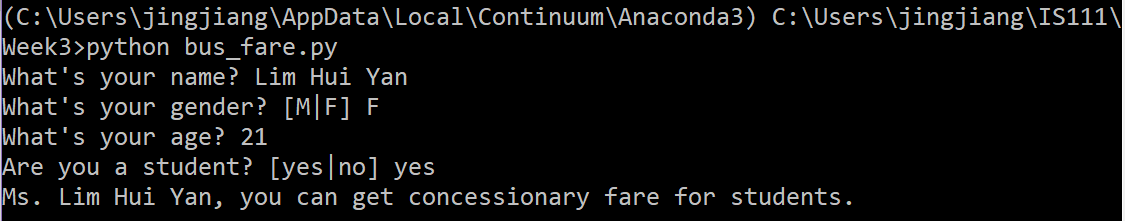
Write code in this file to do the following:

* Use the function get\_user\_info to obtain some information from the user.
* Based on the users’ information, display the following messages accordingly:
  + If the user is 6 years old or below, display "<name>, you can travel for free."
  + If the user is 60 years old or above, display "<Mr.|Ms.> <name>, you can get concessionary fare for senior citizens."
  + If the user is above 6 and below 60, and the user is a student, display "<Mr.|Ms.> <name>, you can get concessionary fare for students."
  + For all the other users, display "<Mr.|Ms.> <name>, you need to pay full fare."

Three sample runs of the program can be found below:







**Q6 [ \*\*\* ]: Taxi Fares**

We are going to implement a taxi fare calculator.

The real taxi fare structure is fairly complex. Refer to the following link for how taxi companies calculate taxi fares in Singapore:

<https://www.lta.gov.sg/content/ltaweb/en/public-transport/taxis%20and%20private%20hire%20cars/fares-and-payment-methods.html>

Because the real fare structure is so complex, we will use the following simplified fare structure for this question.

* The total taxi fare of a ride consists of two parts: (1) meter fare, and (2) surcharges.
* The meter fare of a ride consists of two parts: (1) flag-down fare, and (2) fare based on distance rate.
  + The flag-down fare convers the first **1km** of a ride. If the ride is shorter than 1km, the total meter fare is the flag-down fare.
  + If the ride is between **1km** and **9.8km**, the additional distance beyond 1km incurs a fare based on a distance rate per 400 meters. (Note that we use 9.8km instead of 10km here to simplify the calculation later.)
  + For any additional distance beyond **9.8km**, there is another distance rate per 350 meters.
* We consider two kinds of surcharges: (1) time-based surcharges, and (2) location surcharges.
  + We assume that if the ride is during a peak period, the time-based surcharge is 25% of the meter fare. If a ride is between midnight and 6am, the time-based surcharge is 50% of the meter fare. Other rides don’t have any time-based surcharge.

For example, suppose we have the following rates for a taxi ride:

* Flag-down: $3.50
* Beyond 1km and up to 9.8km: 22 cents per 400 meters
* After 9.8km: 22 cents per 350 meters

Suppose a taxi ride has a distance of 11400 meters. Also suppose that the ride is between midnight and 6am, and there is a location surcharge of $3.00.

Then the total taxi fare is calculated as follows:

* Meter fare:
  + Flag-down for the first 1km: $3.50
  + Next 8800 meters: $0.22 \* 22 = $4.84
  + Next 1600 meters: $0.22 \* 5 = $1.10
  + Total meter fare: $9.44
* Time-based surcharge:
  + $9.44 x 50% = $4.72
* Location surcharge:
  + $3.00
* Total fare: ($9.44 + $4.72 + $3.00) = $17.16

Create a python script called taxi\_fare.py that helps a user calculate the taxi fare of a journey.

Two sample runs of the program can be found below.

**Note:**

* For the distance travelled, you can assume that the user is always going to enter a whole number.
* If the ride is during a peak period, the program should not ask the user whether the ride is between midnight and 6am.

Design function(s) to help you in your implementation.

