The code in this folder (‘assignment2\_ChongWei.py’) helps to recommend a canteen in NTU based on three different search criteria:

* Keyword-based Search: Recommends canteens based on inputted cuisine-based keywords. The keywords can be concatenated using logic words such as ‘and’ or ‘or’ and is able to recognise the logic words regardless of capitalization or trailing and prevailing whitespace. If keywords are separated by whitespace, input is assumed to have logic word ‘and’.
* Price-based Search: Recommends canteens based on cuisine-based keywords and the indicated budget. It uses the keyword-based search to filter the canteens first based on keywords, then shortlists suitable canteens that are within the maximum allowable price. It then returns the canteens in descending affordability.
* Location-based Search: Recommends the nearest canteens to each of 2 users based on their respective locations. The number of canteens to be recommend is indicated by the user.

The criteria for the above searches follow strictly with the guidelines set in the provided assignment document (“RE1016\_Assignment 2.pdf”).

Certain changes that have been made to the original source code include:

* **Addition of various helper functions**: This allows the program to collect and format the data from one form to another, such that the search functions can understand the input data properly and correctly. Error correction and validity checking have also been added to the helper functions. These are the helper functions made:
  + **distance(**x1,y1,x2,y2): Returns a Float equivalent in value to the Euclidean Distance based on two pairs of Cartesian Coordinates using the formula:
  + **findneareststalls**(user\_locations,distancedict,k): Returns a list of the nearest *k* canteen stalls and their respective distances to the midpoint of two users as well as the distance to each user, depending on the location of the canteens as indicated by *distancedict* and the distance of the users provided by *user\_locations*.
  + **andkeyword**(canteen\_stall\_keywords,keywords): Returns a list of canteens and stalls within canteens from *canteen\_stall\_keywords* that match *­keywords* inputted by user that contain logic word ‘and’.
  + **orkeyword**(canteen\_stall\_keywords,keywords): Returns a list of canteens and stalls within canteens from *canteen\_stall\_keywords* that match *­keywords* inputted by user that contain logic word ‘and’.
  + **andprice**(canteen\_stall\_keywords,keywords,max\_price): Returns 3 outputs:
    - A list of canteens with corresponding stalls from­ *canteen\_stall\_keywords* that are within the max price provided by *max\_price* and also match input *keywords* from user (assuming logic word ‘and’ is used)
    - A string of the canteen, stall and the closest price to the max\_price (to be used if no stalls are within max\_price)
    - A list of lists, each list containing 3 strings corresponding to the canteen name, stall name and price of the stall (to be used for price comparison and sorting according to descending affordability)
  + **orprice**(canteen\_stall\_keywords,keywords,max\_price): Returns 3 outputs:
    - A list of canteens with corresponding stalls from­ *canteen\_stall\_keywords* that are within the max price provided by *max\_price* and also match input *keywords* from user (assuming logic word ‘or’ is used)
    - A string of the canteen, stall and the closest price to the max\_price (to be used if no stalls are within max\_price)
    - A list of lists, each list containing 3 strings corresponding to the canteen name, stall name and price of the stall (to be used for sorting according to descending affordability)
  + **compare**(price\_compare): Returns a list sorted by descending affordability.
  + **combinepricelist**(price\_compare): Returns a list of strings of

“Canteen – Name – Price”

You may refer to the comments in the source code, which will explain what each portion of each function does, to achieve its intended outcome.

* Kee Chong Wei (U2320846D)