1. **(25 Points)** Write a complete, working Python 3 program called euidA.py (where “euid” is your EUID) that does the following:

* Write your EUID in comments at the top of the file.
* Define a user-defined function called create\_list() that accepts three arguments, the first one an integer variable called stop, the second one an integer variable called start with a default value of 0, and the third one an integer variable called step with a default value of 1. Inside the function:
  + Using a loop of your choice and the range() function, create a list based on the start, stop, and step values passed to the function. For example, using the default start value of 0, a stop value of 5, and the default step value of 1, you would generate the following list: [0, 1, 2, 3, 4].
  + Once the list is created, return the list back to the calling function.
* Inside the main part of the program:
  + Prompt for and read in an integer for the stop value.
  + Prompt for and read in a string on whether the user wants to enter a start value? As long as the first character of the string is an uppercase 'Y' or a lowercase 'y', you will accept the response as a positive response. This means that if the user types 'y283rsdfn', you will still accept it as a positive response since the first character of the string is 'y'.
  + If the response for a start value is positive, you will:
    - Prompt for and read in an integer for the start value.
    - Now, prompt for and read in a string on whether the user wants to enter a step value? As long as the first character of the string is an uppercase 'Y' or a lowercase 'y', you will accept the response as a positive response. This means that if the user types 'y283rsdfn', you will still accept it as a positive response since the first character of the string is 'y'.
    - If the response for a step value is positive, you will:
      * Prompt for and read in an integer for the step value.
      * Call the create\_list() function, passing in the read in values for stop, start, and step, in that order, assigning the returned list to a variable called my\_list.
    - Otherwise, if the response for a step value is not positive, you will:
      * Call the create\_list() function, passing in the read in values for stop and start, in that order, assigning the returned list to a variable called my\_list.
  + Otherwise, if the response for a start value is not positive, you will:
    - Call the create\_list() function, passing in the read in value for stop only, assigning the returned list to a variable called my\_list.
  + Print out the list called my\_list to the terminal.
* You may assume the user enters all elements using the appropriate data type. Due to time constraints, no further comments are required.

Here is a sample output to help you write the code. The items in bold are entered by the user.

$ **python3 mat0299A.py**

Enter a stop value: **15**

Do you want to enter a start value? (Y/N) **y283rsdfn**

Enter a start value: **4**

Do you want to enter a step value? (Y/N) **n**

[4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

$ **python3 mat0299A.py**

Enter a stop value: **8**

Do you want to enter a start value? (Y/N) **sjdhf**

[0, 1, 2, 3, 4, 5, 6, 7]

$ **python3 mat0299A.py**

Enter a stop value: **24**

Do you want to enter a start value? (Y/N) **Y**

Enter a start value: **3**

Do you want to enter a step value? (Y/N) **Ysdk2**

Enter a step value: **3**

[3, 6, 9, 12, 15, 18, 21]

1. **(25 Points)** Write a complete, working Python 3 program called euidB.py (where “euid” is your EUID) that does the following:

* Write your EUID in comments at the top of the file.
* Define a user-defined function called short\_sentence() that accepts one argument, a string. Inside the function:
  + Generate a random integer between 2 and 4, inclusively, and assign to a variable called min\_length.
  + Print a message that indicates that words less than or equal to the value of min\_length characters will be removed.
  + Split the sentence string into a list of strings, using a space as the separator, and assign to the list variable called word\_list.
  + Iterate through word\_list and discard any word whose length is less than or equal to min\_length. All other words greater than min\_length shall be added to a new list variable called new\_list.
  + Join the list of strings in new\_list together as a single string, using a space as the separator, and assign to the string variable called mod\_sent.
  + Print out the mod\_sent string variable.
* Inside the main part of the program:
  + Prompt for and read in a sentence as a string, including punctuation.
  + Iterate through the string to create a new string variable called new\_sent consisting of only alphanumeric or space characters. Otherwise (if a character is punctuation), you will print a message indicating that the punctuation character is being removed.
* Call the short\_sentence() function, passing in the string variable new\_sent.
* You may assume the user enters all elements using the appropriate data type. Due to time constraints, no further comments are required.

Here is a sample output to help you write the code. The items in bold are entered by the user.

$ **python3 mat0299B.py**

Enter a sentence: **But, for my own part, it was Greek to me.**

Removing: ,

Removing: ,

Removing: .

Words less than or equal to 2 characters will be removed.

Modified sentence: But for own part was Greek

$ **python3 mat0299B.py**

Enter a sentence: **To a great mind, nothing is little.**

Removing: ,

Removing: .

Words less than or equal to 4 characters will be removed.

Modified sentence: great nothing little