# Introduction to Computer Science (ICS 3U)
# List Assignment

1. Write a program that will allow the user to enter 5 strings into an list and output the contents of the list after all of the input has been entered one value per line. Call your program **List1.py**

   **Hint**: Use two for loops - one to load (input the list) and then another to output the list

2. Write a program that will allow the user to enter 5 numbers **or** strings and output the values in reverse order one value per line. Call your program **List2.py**

3. Write a program that will accept **any** number of numbers from a user and store them in an list. It will then add up all the numbers in the list and output the sum as well as the numbers inputted by the user. Call your program **List3.py**

**Note** Make sure you complete the "Predefined Functions Used With Lists" handout and watch the video "Lists In a Menu Driven Program" before continuing past this point.

4. a) Create a program that will allow the user to enter 5 numbers at the beginning of the program. The user will then be presented with a menu and will be able to choose to output one or more of the following for those numbers:
   - the total of the numbers (**Hint**: look up the sum() function)
   - the average of the numbers
   - the highest number (**Hint**: look up the max() function)
   - the lowest number (**Hint**: look up the min() function)
   - the contents of any element (ie. if the user inputted 5, the value of element 5 would be outputted

   Use void functions for each menu item and pass in a list as the formal parameter. Use the predefined functions used with lists whenever possible - see the handout "Predefined Functions Used With Lists". For this program, the list may appear together when outputted (ie. The output format [1, 3, 5] is acceptable). Call your program **List4.py**

   b) Add the following items to your program:
   - output the original list
   - output the number of times a number is in the list
   - allow the user to input a new item to the end of the list (append an item to the list)
   - allow the user to insert a new item anywhere in the list
   - allow the user to delete a value in the list by inputting the value to be deleted
   - allow the user to delete a value in the list by inputting the index of the element containing the value to be deleted

   Re-save your program **List4.py**

   c) (*Level 4*) Add code to make your program as user-friendly and crash proof as possible (ex. ensure the user knows what values are already in the list before deleting it, make sure the element inputted by the user does not create a "out of range" error, etc.)

5. a) Write a program that will accept five names and **sort** them into alphabetical order. Have the words appear one per line. Call your program **List5.py**

   b) Create a program that will accept five names and then give the user the choice of seeing the original list or a sorted list.

6. a) Create a program which will allow the user to input a list of any number of names. This list should then be outputted to the monitor sorted in alphabetical order. Call your program **List6a.py**

   b) Create a program which will allow the user to input a list of any number of names. This list should then be outputted to the monitor sorted in reverse alphabetical order (ie. From z to a). Call your program **List6b.py**

7. *(Level 4)* Write a program that will accept any number of integers and output the median (the "middle" value) of the list. Save your program as **List7.py**

   **Hint:** See the note called "Finding the Median Value In A List of Numbers"

**Note** Make sure you read the note on Related Lists before doing the rest of the program.

8. a) Write a program that will initialize a list of 10 names and a related list of 10 phone numbers. It will then allow the user to input a name and will output the name and corresponding phone number. Output an appropriate message if the name the user inputted is not on the telephone list. Allow the user to continue finding numbers until they wish to exit the program. Call your program **List8a.py**

   b) Write a program that will input a list of 10 names and phone numbers from a single file and load them into two related lists. It will then allow the user to input a name and will output the name and corresponding phone number. Output an appropriate message if the name the user inputted is not on the telephone list. Allow the user to continue finding numbers until they wish to exit the program. Call your program **List8b.py**

9. Create a program which will input a list of 10 names and phone numbers from a file and load them into an list (you can use the same file you created for List8b.py). The program will then output all of the names on the screen preceded by a number. The user will then enter the number beside the name and have the corresponding phone number outputted. Allow the user to continue finding numbers until they wish to exit the program. Call your program **List9.py**

10. a) Write a program that will accept **any** number of words from a user. The program should then capitalize the first and last letter in every word if the letter is not already a capital and sort it alphabetical order then have it outputted. Call your program **List10.py**

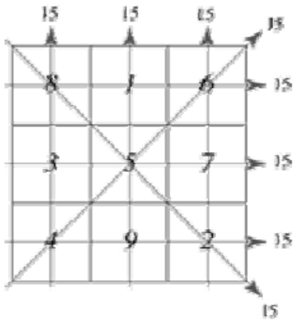
b) Improve your program in (a) to also sort the list in reverse alphabetical order and have it outputted. Re-save your program **List10.py**

c) *(Level 4)* Improve the program in (b) to also sort and output the list in alphabetical order by last letter. Re-save your program **List10.py**

**Note** Make sure you read the note on Two-Dimensional Lists before doing the rest of the program.

11. Create a program that will ask the user to input the number of rows and columns they want in a table. It will then allow the user to input a series of integers and have an appropriately sized table outputted. Call your program **List11.py**

12. a) *(Level 4)* Create a program that will allow the user to input a 3 by 3 table and output whether it is a magic square. A magic square is a table of numbers consisting of the distinct positive integers 1, 2, ..., n^2 arranged such that the sum of the n numbers in any horizontal, vertical, or main diagonal line is always the same number (see diagram on right). Call your program **List12.py**



| | | |
|---|---|---|
| 8 | 1 | 6 |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

b) *(Level 4)* Improve your program so the user can input a table of any dimensions and output whether it is a magic square. Re-save your program as **List12.py**