Project Ideas

The following are a tentative list of project ideas. Students are free to choose their own. For any of the project ideas, feel free to discuss it with any of the TAs or faculty of ES112. Recall that the project is optional for ES112. However, depending the quality of the project, an additional 5% bonus marks may be awarded. **Under any circumstances, plagiarism is not allowed and will result in severe action**. This means, no copying of code from the web or from your seniors. Note that the following descriptions are just suggestions, you are supposed to add new features to your project that go beyond the one line description given below. **Students are free to choose from ES112 list of projects**.

Somewhat easier:

1. Building a canteen management system: where an active account is maintained for each user. Each user should be able to buy food using his/her balance,until it goes negative, and should be able to pay the canteen-wallah to increase the balance.
2. Write a program that plays Tic-Tac-Toe with the user. Ideally, the program should try to win.
3. Lookup the rules for Haiku (short poems) writing on Wikipedia. Write down a program that reads a file containing a large number of English words and then writes a Haiku that uses that word. Of course, it does not really have to have a strict meaning. It is upto you how you want to use the words. You should also use a list of english words stored in a file.
4. Create a login system for a number of machines. The systems should allow adding users, deleting users and converting users to superuser, check user logins etc. It should not store the passwords of the users as such, but should keep some encrypted/hashed version of it that enables checking whether the user has given the right password.
5. Write a program for eight queens puzzle problem of placing eight queens on a chessboard such that no two queens share the same row, column or diagonal. You can also generalize this to **n** queens problem of placing **n** queens on nxn chess board. Write a function to count possible number of distinct solutions.
6. File management system: Options on files for user will be display, create, delete, copy, count, search and modify. Any other interesting operations are also welcome.
7. Use the matplotlib library to draw fractals, or visualize any dynamical system. Choose any of the subset of popular fractal sets.
8. Anagram creator: Use a dictionary of words to find all anagrams of a given phrase. Remember that anagrams have to be valid words.
9. Solve some of the creative exercises from http://introcs.cs.princeton.edu/python preferably from the latter chapters.

Slightly harder:

1. Using the pygame library to build a new game. Pygame is available at www.pygame.org .
2. Visualization of a given dataset. The python library matplotlib will be useful here. Make it interactive by using appropriate clustering algorithms etc.? For appropriate datasets here, contact Anirban.

3. Comparing any two of the sorting algorithms: implement different (known) algorithms to sort a given set of data and then count the number of operations done. Possibly show as a plot the number of operations as a function of the number of inputs.
4. Implement an advanced data structure e.g. a stack or priority queue or heap or balanced trees.
5. Write down a code breaker for the common encryption scheme. For e.g. in the Caesar encryption, we pick a shift and add the value of the shift to each letter of the input text to create the encrypted text. In order to write a code breaker for this, first read in a big piece of text that calculates the frequency of the letters. Then use this information to find out which the most frequent letter is and thus decode.
6. Write down a Sudoku solver, i.e. given as input a description of a (say 4x4) Sudoku board in a file, write down a program that gives a full answer, or tells the users that that the board is not solvable.
7. Implement a class for storing arbitrary length numbers and for doing addition, multiplication and subtractions for these.
8. Implement a program that compares two programs and says whether they are very similar or not. I.e. implement a plagiarism detector. For exact definition, talk to Anirban.
9. Write down a program that reads two books from Project Gutenberg and does a language analysis, by creating a Markov generator for the text. Talk to Anirban about more details.
10. Use some package e.g. https://github.com/bear/python-twitter and visualize the statistics e.g. number of tweets about some prominent Indian political personalities or hashtags in the last few hours.
11. Write a python application to help with PhD application sorting. It needs to i) read csv files ii) show a html form and take feedback iii) store feedback in a database. Talk to Bireswar for more details.
12. Consider the 15 square game (where there are 15 square tiles placed in a 4x4 square, the tiles are movable). Moves only consists of sliding to the empty square. You can represent a board position by a list of lists. Write a program that takes the starting position and enables the user to play the game by inputting moves. You can print out the current board position at every point. You can take the user input of the moves in whatever format suits you best.