

In the following, we will design a SPAM email detection toolbox. First we will compose the emails, then we will detect spam. Do not hardcode values/logic when not specified.

Part A

1. Create a text file named *student_name_list.txt* of the names of 10 PC503 students manually, keeping one name in each line. Note: this is the only file that is to be created manually. Files in the subsequent questions must not be created manually.
2. Write a Python code *main_data.py* to do the following tasks. Each task must be written in a function with a suitable function name. Depending on the task, the function may or may not return anything.
3. Read the file *About-DAIICT.txt* and create four new files named *History.txt*, *Environment.txt*, *Recognition.txt* and *Accreditation.txt* by inserting the content of each section as given in *About-DAIICT.txt*.
4. Create a list named *student_name_list* of the 10 names by reading the text file *student_name_list.txt*.
5. Randomly generate 15 files with file names: *email-1.txt* to *email-20.txt*. So, the name of the 13th file need not be *email-13.txt*, it can also be *email-19.txt* or *email-5.txt*. Content of each file (10 lines) should be as follows:
 - a) First line should have time, date, month, year, email Id in the following format:
Received at 17:23 hrs on October 7, 2018 from fname_lname@daiict.ac.in
All entries must be randomly inserted and email Ids can be created from random elements of *student_name_list*. Obviously, there will be at least one or more files containing the same email Id since the number of students is 10 (<15).
 - b) Second line should be blank.
 - c) Third line to the tenth line must be the lines from the four above mentioned text files in random order such that (i) there is at least one line from each file and (ii) the lines taken from *History.txt* can include only one of the four sentences containing “first wave”, “second wave”, “third wave”, “fourth wave”. There is no restriction on other sentences in this text file.
6. Randomly generate 4 more files using the above mentioned logic with two modifications:
 - a) File name should be the unused numbers between 1 to 20.
 - b) Email Ids must be taken from the following list: email_ids = [name1@gmail.com, A_X_y@yahoo.co.in, nm123@rediff.com, nam_4_e@160.com].
7. Randomly generate 1 more file using the above mentioned logic with three modifications:
 - a) File name should be the unused number between 1 to 20.
 - b) Email Id is pc_503@daiict.ac.in
 - c) The lines taken from *History.txt* cannot include any one of the four sentences containing “first wave”, “second wave”, “third wave”, “fourth wave”. There is no restriction on other sentences in this text file.

Part B

8. Write a Python code *main_test.py* to do the following tasks. Each task must be written in a function with a suitable function name. Depending on the task, the function may or may not return anything.
 9. Print the email Ids whose files do not have any “wave” information. In the current exercise, there is only one such entry.
 10. The users not having daiict email Id or without any “wave” information must be spammed. Rename the corresponding files as *spam1.txt*, *spam2.txt*, ... and update their contents by adding a line at the beginning: “This email has been categorized as spam”.
 11. Create *ordered_names.txt* containing the user names (non-spam type) in the descending order of date-time etc., i.e. most recent first. One name in each line. Keep duplicate names.
 12. Create *ordered_names_wave1.txt* containing the user names (non-spam type) belonging to the “first wave” group in the descending order of date-time etc., i.e. most recent first. No duplicate names and one name in each line. Similarly create files for the “second wave”, “third wave”, “fourth wave” groups.
 13. Plot the number of users in each group and number of spam emails using a bar chart.
- Note:** Randomize the codes so that every time they are executed, different output is expected.