

## LAB1:Python Exercise

Python is excellent for text processing and functional programming, as we'll find out in the course of these exercises. Many exercises will use the following (imagined) context.

GoldMan Sachs was one of the companies which came for placement to SGSITS, Indore. Their selection process included five written tests followed by an interview. The written tests had up to 5 marks each. Students could decide not to attempt a written test, since if they get the written test totally wrong, they may get negative marks.

The file marks.ods gives the spreadsheet of marks obtained by each of the candidates who appeared for selection. There are some heading rows in the spreadsheet. The first row before the actual data in the spreadsheet gives the column names which stand for the various written tests. In the cells, a "NA" means that the candidate did not attempt that particular written test (for fear of getting negative marks). The column named "Sel" tells whether the candidate was finally selected after the interview: this is a binary value (1=selected, 0=not selected)

A non-python exercise for you is to first convert this open-office spreadsheet into a csv (comma-separated value) text file called marks.csv.

1. Write a python script to read the csv file and print the total number of candidates who appeared for selection. You must look for a number as the first entry in a row, and count such rows to arrive at the answer. Your program should take exactly one command-line argument, the input csv file. If the wrong number of arguments are given, the program should exit after printing the correct usage.[Outcome:command-line arguments, functions, if-then-else, loops, the csv module, regular expression matching]
2. Modify the above program to store the main data (i.e. excluding the header rows) as a list of lists (2D array). And then print the number of candidates by using "len".  
[Outcome: lists, list-of-lists, the len function]
3. Modify the above program to make a list of only the final selection. Use map on the list of lists to achieve this. Then print the total number of selected candidates using the "sum" function on this list.  
[Outcome: using map on a list]
4. Modify the above program to make another 2D matrix. This should consist of the rows corresponding to only the selected candidates. Use the filter function to achieve this. Then print the total number of selected candidates by using "len" on this 2D matrix.  
[Outcome: using filter on a list]
5. Now add to the above program to make a "dictionary" or a "hashtable" which maps the column name to the column index. To construct this index, you have to use the header row, which you can identify in the program as having "SNo" as the first column value. The program should then loop through the dictionary to print the column index of each column name.  
[Outcome: constructing and looping through dictionary or hashtable structures in python]