

# CO222: Programming Methodology - Project 2

## Meeting Data Visualizer - Specification

### 1 Introduction

Due to the social distancing and travel restrictions all around the world, people have moved most of the work to virtual environments. Due to this, the usage of meeting tools has exponentially increased so as the data gathered through such tools. The objective of this project is to analyze such data files and visually represent the data as per the requirement of user.

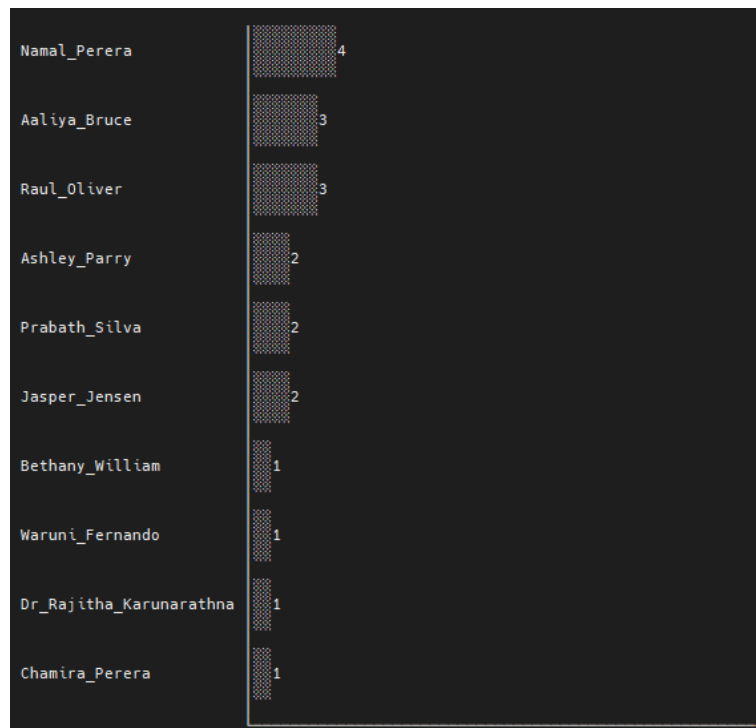


Figure 1: The expected output from the program. People with most number of meetings are displayed as a horizontal bar chart

Fig 1 shows the expected output from the program concerning the maximum number of meetings. There are different control and input arguments for the program. According to the arguments, the program should be able to change its behaviour and result in the expected output.

### 2 Program output

#### 2.1 Control arguments for the program

##### File name/ File Names

The program should be able to accept any number of file names in any order. Input files should be CSV files. Also the file names will not start with '-'. eg: -file.csv

## Number of rows in the chart

The argument specifies the number of rows in the bar chart. It should be given as `-l 10` where 10 is the limit. It can be any positive integer. A number should always follow the `-l` argument. The pair can be in any place of the arguments list.

## Scaled option

When `--scaled` argument is given, the first row of the graph should fully occupy the max print width. Any other row should scale to be matched with first row scale factor.

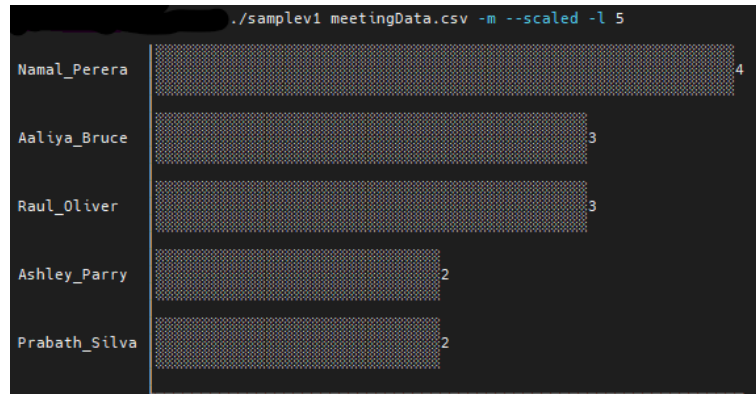


Figure 2: The expected output from the program. 5 people with most number of meetings are displayed as a horizontal bar chart. The bar for the greatest value expands trough out the print area and others are proportional to that

## Meetings/participants/duration Representation

The program can analyse three different parameters, No. of Meetings, No. of Participants and Duration. If the output should list meetings, the `-m` should be given. If the output should list participant counts, the `-p` should be given. If the output should list time duration, the `-t` should be given.

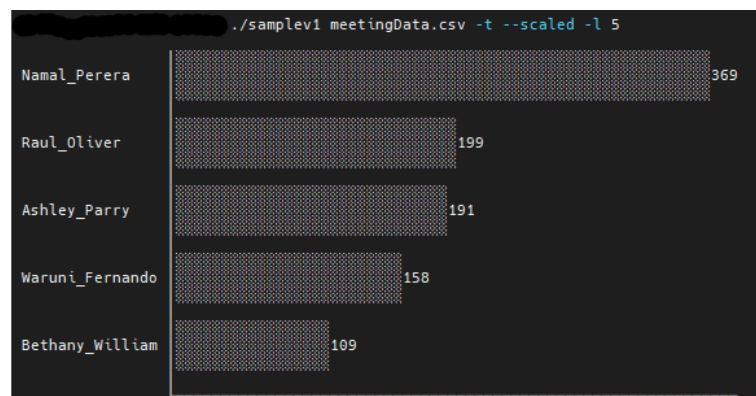


Figure 3: The expected output from the program. People with maximum meeting duration in minutes are displayed as a horizontal bar chart

## 2.2 Default options

The program must take at least one file name to work. All other arguments are optional. If not given, the program will work as **non-scaled**, will output counts for **meeting** and limit the output rows to **10**.

## 2.3 Introduction to data

Each row of the csv file represents one meeting. Each row contains the name of meeting host, number of participants and the duration of the meeting in the format of hh:mm:ss.

While printing, if two entries share the same frequency, the first occurred meeting in the file should be printed first on the chart.

## 2.4 Printing area

The program should work in 80 character width screen. To understand the printing pattern, please refer to the given binary file and test with different files. It will give you a clear understanding about how the graph is printed on the screen. The output should print exactly at the same place and scale as the given program.

You should use `std=c99` flag to compile the source code because there are several Unicode characters you have to use when printing the graph. They are; 2500, 2502, 2514, 2591. It is up to you to find out what exactly these Unicode print. To print Unicode you may use `printf` as follows, `printf("\u2502");`

# 3 Breakdown of the marks

## 3.1 Basic functionality - 50%

If the program can read multiple files, store meeting data, duration data or participant data as given by the user and then produce the maximum N (given by user, default 10) number of entries, then the program will be given 50 marks (even without a graph).

## 3.2 Plotting the chart - 20%

If the graph is plotted with correct output and as expected, the program will be given another 20 marks.

In all of the above cases, you may use the following static pattern of the command line arguments to run the program.

```
./prog -m - -scaled -l 10 file1 file2 file3 ...
```

where, -m can be changed to either -t or -p and 10 can be any positive integer

## 3.3 Input arguments and error handling - 30%

As you can see, the program has many arguments to be processed, and they may appear at any place in the argument list. If your program is capable of handling arguments as the example binary you are given, you may score 30 marks more.

## 4 Submission

Submit a single .c file rename it as the following pattern where xxx is your registration number. **18xxxProject2.c** The project will be auto-marked, please make sure that your program replicates the same functionality as given sample program.

## 5 Important

Under no circumstance, you should copy somebody else's code. Copying someone else's code or showing your source code to anyone else will earn you zero mark for the whole project. (**Hope you got experience from your Project 1 and Labs.**) Therefore, put some honest effort to earn the marks for project 2.

## 6 Deadline

The deadline for the submission is **08th October 2021 23:55h**. (For each late day you'll loose 25% from the mark you obtain.)