GDC Hackathon

There are 5 questions with marks allocated to each. Attempt the questions to the best of your ability. The team with the highest mark at the end of the hackathon win.
Note that all questions are subject to verification of the hackathon panel.
At the end of 3 hours, all teams will be required to stop and the judging will begin.

1. **GDC Forum**
   Our community needs a forum. The features and aesthetics of the forum are at your discretion. We probably need a forum where we can all have profiles and possibly share posts. It will aid collaboration across companies but posts from the forum should be open to the general public.
   Build this forum application (mobile, web or both)

   50 points

2. **Oil Paintings**
   There has been a lot of hype around oil paintings in Nigeria suddenly. We wish to create an automatic painter.
   We figured that the easiest way to do this is to design a machine that responds to commands when it is given an input of a string representation of the picture.
   The picture is a rectangular grid of square cells, each of which either must be painted, or must remain clear. At the beginning, the canvas (all cells) is clear.
   The cells of the picture are referred to using their coordinates. [X , Y] denotes a cell in the X -th row and the Y -th column of the picture. Indexing of the rows and columns is 0-based, with the cell [0, 0] located in the top-left corner of the picture.
   The intended machine supports the following commands:
   - **a.** PAINT_SQUARE X Y S - paints all cells within the square of $(2S + 1) \times (2S + 1)$ dimensions centered at [X , Y]. In particular, the command “PAINT_SQUARE X Y 0” paints a single cell [X , Y]. For the command to be valid, the entire square has to fit within the dimensions of the painting.
   - **b.** PAINT_LINE X1 Y1 X2 Y2 - paints all cells in a horizontal or vertical line between [X1 , Y1] and [X2 , Y2], including both ends, as long as both cells are in the same row or column or both. That is, at least one of the two has to be true: X1 = X2 or/and Y1 = Y2 .
   - **c.** ERASE_CELL X Y - clears the cell [X , Y].

   Input Data
   The input data is provided in a plain text file containing exclusively ASCII characters with lines terminated with a single ‘\n’ character at the end of each line (UNIX- style line endings).
   The file consists of:
   - one line containing the following natural numbers separated by single spaces:

  - N denotes the number of rows of the picture ($0 < N \leq 1000$)
  - M denotes the number of columns of the picture ($0 < M \leq 1000$)
- N subsequent lines describing individual rows of the picture. Each i-th ($0 \leq i < N$) such line contains M characters describing picture cells in consecutive columns of the i-th row of the picture, starting with the column 0. Each character is either:
  - '.' - denoting a cell that has to remain clear
  - '#' - denoting a cell that has to be painted

Input example

5 7 ....#.. ..###.. ..#.#.. ..###.. ..#....	5 rows, 7 columns. First row of the picture. Second row. Third row. And so on. Further redundancy is redundant.

Submissions

For the solution to be accepted, it has to meet the following criteria:

- The format of the file must match the description below
- All cells referenced in the painting commands must fit within the dimensions of the grid
- The provided list of commands must produce the picture specified in the input file. If the resulting picture differs from the target one, the solution will not be accepted.
- The number of painting commands provided cannot be bigger than the number of cells in the picture

File format

A submission file has to be a plain text file containing exclusively ASCII characters with lines terminated with either a single '\n' character (UNIX -style line endings) or '\r\n' characters (Windows- style line endings).

The file has to start with one line containing a single natural number S representing the number of painting commands ($0 \leq S \leq NM$) . Then individual painting commands have to follow, each in a separate line. Each painting command has to conform with the format described in the problem description above.

Example

The following is a valid submission for the input file presented above.

`4`
`PAINT_SQUARE 2 3 1`
`ERASE_CELL 2 3`
`PAINT_SQUARE 0 4 0`
`PAINT_SQUARE 4 2 0`
	4 commands in total.
	Paint a square of 9 cells centered at [2, 3].
	Unpaint a single cell in the middle of the square.
	Paint a single cell at [0, 4].
	Paint a single cell at [4, 2].

The following is also a valid submission for the input file presented above.

`4`
`PAINT_LINE 0 4 3 4`
`PAINT_LINE 1 2 4 2`
`PAINT_LINE 1 3 1 3`
`PAINT_LINE 3 3 3 3`
	4 commands in total.
	Paint a vertical line from [0, 4] to [3, 4].
	Paint a vertical line from [1, 2] to [4, 2].
	Paint a single cell at [1, 3].
	Paint a single cell at [3, 3].

Task and Scoring

The task is to produce the output for the painting in this input (http://bit.ly/gdc-2). All attempts that create this painting fetch you 20 Points, A solution that is optimal in terms of no of instructions fetches an additional 5 marks. A solution that is optimal in terms of time complexity of the code fetches 5 marks.

(Question courtesy Google) **30 points**

3. **The download manager problem**

The ideal download manager works the following way. It sends a http request for a file with headers that request for ranges that split the file so that the download can commence at different sections concurrently on individual threads. Once the downloads are completed, the download manager merges the individual files and "voila", your file has been downloaded.

Here are 11 parts of a binary file (http://bit.ly/gdc-3) saved in a zip file. Your task is to merge these parts to produce a file. The name of the file has been persisted in the parts.

When done merging, show your file to an administrator and voila, you get 10 points.

10 Points

4. **The graph**

Here is a zip file with the json file of JSON objects accounts.json. It denotes bank accounts for several account holders. Represent the data graphically using any graph of your choosing. You may use pie charts, bar charts, line charts, doughnut chart and any fields of the data in your chart(s).

What we want to see is your ability to represent data to give on-the-spot insights.

10 points

May the ForceX be with you