

COL 100M - Lab 10

Week of April 1, 2018

1 Instructions

1. Use the OCaml top-level to develop and debug your code.
2. Please write the code in a file first and then test your code in the OCaml top-level using the directive `#use "foo.ml"` for the file `foo.ml`
3. You may assume that all inputs are valid unless otherwise stated in the problem.
4. In questions that require string outputs, be careful not to include leading or trailing whitespace.
5. You may submit and evaluate your code a *maximum* of 15 times without penalty. Subsequently, you will lose 2 marks per additional evaluation. Therefore, please ensure that you have thoroughly debugged your code before submitting.

The following submission file is *required*:

1. `Linear.ml`
2. The implementation should be done using arrays and loops only. No marks will be given if list or recursive functions are used.

2 Assignment

2.1 Functionality

This week you will make a file `Linear.ml` which computes the solution to a system of Linear equations $Ax = b$. Given a matrix A of size $m \times n$, and a vector b of size m as input (both Valid),

- print "no", if no solution exists
- print "inf", if infinite solution exists
- print the list of unique values, if unique solution exists

For floating point arithmetic, consider all numbers less than epsilon to be zero, where

$$\epsilon = 10^{-10}$$

You have to read the input from a file provided in the standard input and write the output in another file mentioned in the input file.

Note: The implementation should be done using arrays and loops only. No marks will be given if list or recursive functions are used.

2.2 How your program will be run

Your program will be run from the *command line* after compiling it into an executable. That is, your program will be compiled using the following command: `$ ocamlc -o linear Linear.ml`. The executable `linear` is then run(for example) using the command: `$./linear input.txt output.txt`, where `input.txt` and `output.txt` are text files whose format is described in the next section. Note that, your program should consist of a *main* function that will read the contents of the `input.txt` and then perform the operations as specified, and finally write to `output.txt`

2.3 Input

The input to your program will be a file whose name will be specified as a command-line argument, that will consist of multiple lines, according to the following format:

1. From the *first* line onwards, the matrix $[A \ b]$ matrix will be specified in a row-wise manner until you encounter the line with "next" written in it.
2. Each row of matrix will be represented in a space separated format. **The data will be floating point numbers.**
3. Similarly, you will fetch matrices one by one and last line of the input will be the one with "next" written in it.

2.4 Output

Your program should write its results to the output file specified as the second argument given in the command. For each matrix, provided in input file there should be one line in output file, corresponding to the solution of that matrix. In case of unique solution, the solution should be in a space separated format as specified below.

Sample Input File

```
1 -2 3 7
2 1 1 4
-3 2 -2 -10
next
1 2 3 6
2 4 6 6
1 1 1 3
next
1 1 1 3
2 2 2 6
next
```

Corresponding sample Output File

```
2 -1 1
no
inf
```