

Mini Project 1: MATLAB Introduction (3 pages)

Rule 1. Submit an M-file named `mp1_ABC_WXYZ.m`, where `ABC` is your section number and `WXYZ` are the last 4 digits of your NJIT ID. Observe capitalization. Use an underscore `_` instead of a dash `-` or MATLAB will get confused. **SUPPRESS output for all of your MATLAB commands.**

Rule 2. Send an email to the instructor `alexg+cs101@cs.njit.edu`, to the grader `js87+cs101@njit.edu`, and if you want to make sure that everything went fine with the transmission of your mail, to yourself as well. Include the three words `mp1 ABC WXYZ` in the subject line, separated with space if your id ends with `WXYZ` and you are enrolled in CS101-ABC.

Rule 3. It is imperative that you fully conform to Rules 1 and 2. Testing will be done by running a MATLAB program (that will also be given to you after submission). If you deviate from these rules you will be getting errors (and points deducted). Observe variable names and capitalization.

Due Date: Before **noon time of Friday February 28, 2014.**

1 Part A: Warm-up (14 points)

You will create a text-based M-file named (Rule 1) `mp1_ABC_WXYZ.m` where `ABC` and `WXYZ` are as specified in Rule 1. **Use semicolons to suppress output.**

1. First line. The first line of the MATLAB file will contain in the form of a MATLAB comment the name of the file in question i.e. `mp1_ABC_WXYZ.m` with the first character, the `m`, separated by two spaces from the MATLAB comment symbol.

2. Second line. The second line would contain in the form of a MATLAB comment line your last name fully capitalized followed by your first or other given names in lower case (all characters). Then include the last four digits of your id, eg `WXYZ`. Exactly one space character inbetween any two of them or the comment indicator of MATLAB.

3. Third-Seventh lines. The third line will be empty. For 3.1-3.4 use 4 lines (fourth through seventh). The remaining lines of the MATLAB file are described in the following questions 4-25. Pay attention to the details. Variables have names starting with a `v` followed by a number. Do not change names or capitalization. Grading will be done automatically for most problems. **Use brief MATLAB array operators rather than explicit listing of matrix or vector elements. Thus If you need to create a row vector `v` containing `1,2,3,4`, a solution that is along the lines `v= 1:4` is a correct one; writing something like `v = [1 2 3 4]` will get you 0 points. You have been warned!**

3.1 Variable v1. Create variable `v1` and assign to it your last name in lower case in the form of a string. Use one (separate) line.

3.2 Variable v2. Create variable `v2` and assign to it the last four digits of your NJIT id in the form of a string. Use one (separate) line.

3.3 Variable v3. Create variable `v3` and assign to it the last four digits of your NJIT id in the form of a 16-bit integer. Use one (separate) line.

3.4 Variable v4. Create variable `v4` and assign to it the last four digits of your NJIT id in the form of a 16-bit unsigned integer. Use one (separate) line.

2 Part B: Playing with numbers and vectors (28 points)

There are no variables `v5` through `v13`. The next variable to be created will be `v14`.

4. Create a MATLAB variable `v14` that is assigned the value `141.4`.
5. Create a MATLAB variable `v15` that is assigned the value `151.51`
6. Create a MATLAB variable `v16` that is assigned the value $\log(8e\pi)$, where π is the ratio of the circumference to the diameter of a circle, and e is Euler's number.
7. Create MATLAB variable `v17` that is assigned the value 17×10^7 .
8. Create MATLAB variable `v18` that is assigned the value $2^\pi/(4 * e^2)$. (see 6 above for definitions of π , e .)
9. Create MATLAB vector variable `v19` that is assigned the values $1, 1/2, 1/3, 1/4, 1/5, \dots, 1/10000$ in that order (it is a sequence of 10000 terms).
10. For vector `v34= [1:100]`, create MATLAB variable `v20` that adds 9 to each one of its 100 elements.
11. For the same vector `v34= [1:100]`, create a MATLAB variable `v21` that adds 3 to each of its odd indexed elements.
12. For the same vector `v34= [1:100]`, create a MATLAB variable `v22` that computes the square of each one of its elements.
13. For the same vector `v34= [1:100]`, create a MATLAB variable `v23` that computes the square-root of each one of its elements.
14. Using the colon operator create row vector variable `v24` such that

$$v24 = [24 \ 20 \ 16 \ 12 \ 8 \ 4 \ 0].$$

15. Using the colon operator as needed, create vector variable `v25` such that

$$v25 = \begin{bmatrix} 3 \\ 7 \\ 11 \\ 15 \\ 19 \end{bmatrix}$$

16. Create row vector variable `v26` such that it includes all the numbers from 10 to -10 in step increment of -0.5 so that

$$v26 = [10.0000 \ 9.5000 \ 9.0000 \dots - 9.0000 \ - 9.5000 \ - 10.0000].$$

17. Create vector variable `v27` such that it becomes a string of length 23 whose value is shown below. The string includes one space between two words, and the character following `fallin` is a quotation mark.

$$v27 = \text{I am fallin' for MATLAB}$$

3 Part C: Playing with matrices (18 points)

18. Create vector variable `v28` such that it becomes a string of 5 characters containing NJIT President's last name. The first character should be in capital and the remaining ones in lower case.

19. Create a matrix variable `v29` with 5 rows each row containing NJIT President's last name. The first character should be in capital and the remaining ones in lower case.

20. Create a MATLAB variable `v30` that is assigned the value of the number of 1s of matrix `ones(53,53)+eye(53)+ zeros(53,53)-1`.

Use only vector operations do not input matrix elements explicitly. You may define auxiliary variables starting with `abcd_WXYZ_` and then using a single digit of your choice: be remind that `WXYZ` are the last four digits of your ID. You will be slightly penalized for the use of such variables however. You are not allowed to use `reshape`, if you are tempted to.

21. Create matrix variable `v31` such that it is a 31×31 matrix of 21s. (You are not allowed to type 21 or 31 more than once.)

$$v31 = \begin{bmatrix} 21 & \dots & 21 \\ \vdots & & \vdots \\ 21 & \dots & 21 \end{bmatrix}$$

22. Create matrix variable `v32` such that it is a 10×10 matrix of the numbers from 1 to 100 in column major order as shown below. (You may not use more than four times the colon operator.)

$$v32 = \begin{bmatrix} 1 & 11 & \dots & 91 \\ 2 & 12 & \dots & 92 \\ \vdots & \vdots & \vdots & \vdots \\ 9 & 19 & \dots & 99 \\ 10 & 20 & \dots & 100 \end{bmatrix}$$

23. Create matrix variable `v33` such that it is a 10×10 matrix of the numbers from 1 to 100 in row major order as shown below. (You may not use more than four times the colon operator.)

$$v33 = \begin{bmatrix} 1 & 2 & \dots & 10 \\ 11 & 12 & \dots & 20 \\ \vdots & \vdots & \vdots & \vdots \\ 91 & 92 & \dots & 100 \end{bmatrix}$$

4 Part D: Grading

Most (but not all) of the grading would be performed by a program `mp1_104_chk.m` which is an M-file. When you have access to it (the day after the submission deadline), open a matlab command window, first perform a `clear` to reset your workspace. Then run the M-file of your submission `mp1_ABC_WXYZ.m` as needed, and then run our script by typing `mp1_104_chk` on a new MATLAB command prompt. Our M-file does most but not all of the grading: it won't check lines 1-5. Someone else will do it! But your actual grade would be within 10 points of the reported grade.