

Mini Project 1: MATLAB Introduction (4 pages)

Rule 1. Submit per Handout 2 guidelines, at least `mp1_ABC_WXYZ.m`, where `ABC` is the section number and `WXYZ` the last 4 digits of your NJIT ID. Observe capitalization. Use an underscore `_` instead of a dash `-`; **SUPPRESS output for all of your MATLAB commands.**

Rule 2. Send an email (with subject line that includes the three words `mp1 ABC WXYZ`, separated with one space, as applied to you)

(a) to the instructor `alexg+cs101@njit.edu`,

(b) to the grader whose email address will be posted after Sep 20 on the course Web-page, and

(c) to make sure that everything went fine with the transmission of your mail, to yourself as well.

Rule 3. It is imperative that you fully conform to Rules 1 and 2. Testing will be done by running a MATLAB program; If you deviate from these rules points will get deducted. Observe variable names and capitalization.

Due Date: Before noon time of Tue Sep 30, 2014.

1 Part A: Warm-up (15 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. **Always 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 em, separated with three (3) 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 three (3) space character inbetween any two of the words 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 4-26. 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.**

3.1 Variable v1. Create variable `v1` and assign to it your last name in capital (upper) 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 32-bit unsigned 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 32-bit integer. Use one (separate) line.

2 Part B: Playing with numbers and vectors (42 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 `141.41`.
5. Create a MATLAB variable `v15` that is assigned the value `151.5` and be of datatype `single`.
6. Create a MATLAB variable `v16` that is assigned the value $\log(6e^6\pi^{1/6})$, 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 1.7×10^7 .
8. Create MATLAB variable `v18` that is assigned the value $18^\pi / (18 * e^{18})$. (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, ..., 1/1000` in that order (it is a sequence of 1000 terms).
10. Create vector `v34= [1:2:100]`, and create MATLAB variable `v20` that subtracts 1 from each one of its elements.
11. For the same vector `v34= [1:2:100]`, create a MATLAB variable `v21` that adds 10 to each of its odd indexed elements.
12. For the same vector `v34= [1:2:100]`, create a MATLAB variable `v22` that computes the square of each one of its elements.
13. For the same vector `v34= [1:2:100]`, create a MATLAB variable `v23` that computes the cubic-root of each one of its elements.
14. Use the colon operator to create row vector variable `v24` such that

$$v24 = [24 \ 21 \ 18 \ 15 \ 12 \ 9 \ 6 \ 3].$$

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

$$v25 = \begin{bmatrix} 30 \\ 70 \\ 110 \\ 150 \\ 190 \\ 230 \end{bmatrix}$$

16. Create row vector variable `v26` such that it includes all the numbers from 10 to -10 in step increment of -0.25 so that

$$v26 = [10.0000 \ 9.7500 \dots \ -9.7500 \ -10.0000].$$

17. Create vector variable `v27` such that it becomes a string of length 27 whose value is shown below. The string starts with the I, has one space between two consecutive words, ends with exclamation marks.

$$v27 = \text{I like MATLAB a lot!!!!!!!!}$$

3 Part C: Playing with matrices (28 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 6 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 `2*ones(330,330)+3*eye(330)-4`.

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 31s. (You are not allowed to type 31 more than twice.)

$$v31 = \begin{bmatrix} 31 & \dots & 31 \\ \vdots & & \vdots \\ 31 & \dots & 31 \end{bmatrix}$$

22. Create matrix variable `v32` such that it is a 10×10 matrix of the numbers from 10 to 1000 step 10 in column major order as shown below. (You may not use more than four times the colon operator.)

$$v32 = \begin{bmatrix} 10 & 110 & \dots & 910 \\ 20 & 120 & \dots & 920 \\ \vdots & \vdots & \vdots & \vdots \\ 90 & 190 & \dots & 990 \\ 100 & 200 & \dots & 1000 \end{bmatrix}$$

23. Create matrix variable `v33` such that it is a 10×10 matrix of the numbers from 10 to 1000 step 10 in row major order as shown below. (You may not use more than four times the colon operator.)

$$v33 = \begin{bmatrix} 10 & 20 & \dots & 100 \\ 110 & 120 & \dots & 200 \\ \vdots & \vdots & \vdots & \vdots \\ 910 & 920 & \dots & 1000 \end{bmatrix}$$

24. You have already created matrix variable `v34`, haven't you ?

25. Create matrix variable `v35` such that it is initialized to the row-vector `v24`, and the use a single assignment statement to change the following three of its values:

$$v35(1) = 10 \quad , \quad v35(3) = 40 \quad , \quad v35(5) = 70 \quad ,$$

26. Create matrix variable `v36` such that it is set to the sum of the values of the elements of `v19` minus the log of the number of elements of the vector `v19`. The result is an approximation to the value of Euler's constant γ which is approximately 0.5772156649..... (do not expect more than 2-3 decimal digits of precision though...)

4 Part D: Grading

Most (but not all) of the grading would be performed by a program `mp1_000_chk.m` which is an M-file. When you have access to it (noon Friday after the Tuesday submission), open a matlab command window and then 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_000_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. The grader will do it! But your actual grade would be within 10 points of the reported grade.