

**University of Massachusetts Lowell
Electrical and Computer Engineering
EECE 1070 Intro to Engineering for ECE
Writing Programs and functions in Matlab**

Date: _____

Section # _____

Learning Objectives:

By completing this Lab exercise you should be competent in the following learning objectives:

- Getting more experience with Matlab functions
- Using commands: while, if then – elseif to control program structure
- Writing simple programs in Matlab

Instructions:

Part A. Grading Problem, Using Control Structures

In 25108, all the Labs, MPs and exams consist 600 points. The final letter grade is given according to the following table:

E	0-300
D-	300-320
D	320-340
D+	340-360
C-	360-380
C	380-400
C+	400-420
B-	420-450
B	450-480
B+	480-510
A-	510-540
A	540-570
A+	570-600

In this section, the work is divided into 3 parts: a main program “Convert” that allows you to input the points a student earned, a function “Grade” which decides which letter grade the student should get, and another function “plus_minus” which decides whether the student should get + or –.

The outlines of these 3 parts are given below, you should fill in your code to make them run.

Convert.m

```
...  
Points = input ( ... ); % ask user to input the points a student earned  
Finalgrade = Grade(Points); %calls function Grade to decide the letter grade  
...
```

Grade.m

```
% Grade function only decides the letter students should get  
% It calls plus_minus to decides what goes behind, a “+” or a “-”
```

```
Function g = Grade (points)  
If points >= 510  
g = [‘A’, plus_minus(points, 540, 570)] %Here a concatenation is used  
...
```

plus_minus.m

```
% this function returns a “+” or a “-” or nothing.  
% this function takes 3 parameters: the points a student got, a low point, a high point.  
% when the points is higher than high point, the student gets a “+”  
% if the points is lower than low point, he gets a “-”.
```

```
Function pm = plus_minus(points, lpoint, hpoint)  
If ....  
pm = “-”;  
...
```

Example: Steven got 520 points, now we want to know his letter grade. First we run the program “Convert” in Matlab command window. The output is:

Convert

Please input the points the student earned: 520

A-

What will happen if you input 800 or -20 as the points a student got?

Convert

Please input the points the student earned: 800

How do you solve this problem?

Part B. In this section, you are going to write a function that can add two polynomials of different degree.

1. Given two polynomials: $P = x^3 + 2x^2 + 3x + 1$, $Q = x - 1$.

How to represent them? $P =$ _____, $Q =$ _____.

What will happen if you want to add them up using $P+Q$?

2. We cannot sum two vectors with different dimensions. We have to make their dimensions the same by adding 0's to Q. The function `zeros(a,b)` can help.

Try commands `zeros(2,3)` and `zeros(1,4)`.

What's the result of function `zeros(a,b)`? _____

Now write down a command that change Q's degree to the same as P, without changing Q's value.

$Q =$ _____

3. Refer to notes for the use of function `size(p)`.

Now suppose we have two polynomials R and S, we only know that R has a lower degree than S. By using a combination of commands `size` and `zeros`, we can change R's degree to the same as S's.

`Vec = (size(R) - size(S)) %` get the difference between R and S. If this is positive you need to add zeros to S, if negative you need to add zeros to R.

```
% vec(2) will be the difference.
R = [ zeros(_____), R] % concatenate zeros and R
```

4. Write a function adding two polynomials of different degrees and give out the sum as result.

For example, your function should behave as follows:

```
Add(P,Q)
```

```
ans =
[1,2,4,0]
```

Hint: your function should be able to sum up any two given polynomials, so you may first decide which polynomial has a higher degree. Use the results of step2 and step3.

Part C. Chebyshev polynomials.

Chebyshev polynomials are defined as follows:

$$T_0(x) = 1;$$

$$T_1(x) = x;$$

$$T_n(x) = 2*x*T_{n-1}(x) - T_{n-2}(x)$$

Note: multiplying any of our polynomials by “x” is in effect shifting the polynomial to the left and adding a 0 in the right most coefficient.

Write a function "Cheby(n)" that returns the Chebyshev polynomial for a given n in MATLAB vector polynomial format.

For example, your function will do the following:

```
Cheby(2)
```

```
ans=
```

```
[2, 0, -1] for  $T_2 = 2x^2 - 1$ 
```

The structure of the function is given below, you should fill in necessary code to make it run.

```
Function c = Cheby(n)
```

```
c0 = [1];
```

```
c1 = [1,0]; % initialization for Cheby(0) and Cheby(1)
```

```
i = 1; % initialization for loop control variable used in while loop
```

```

if n = 0
    _____;
elseif n = 1
    _____;
else
    ci2 = c0; % ci2 always represents Cheby(i-2)
    ci1 = c1 % now i = 1, ci1 always represents Cheby(i-1)

% in the following loop, the program always computes Cheby(i)
% as i increases, all the Chebyshev polynomials Cheby(2), Cheby(3)...are calculated
% when i hits n, Cheby(n) is calculated.

while i < n
    i = i + 1;
    ci = _____; % now i increases, we calculate
    the
    % new cheby(i) accordingly.
    % Hint:  $c_i = 2x c_{i-1} - c_{i-2}$ 
    % you should utilize the “add” function implemented in Part B
    % and to multiply  $c_{i-1}$  with  $2x$ , multiply by 2 and append a zero at the right end.
    ci2 = ci1;
    ci1 = ci;
end

```

Check your answer by computing cheby(5) and cheby(10)

Hand in you code for each of the 3 parts to your TA. This lab counts for 200 points.