Sample of the First Midterm Exam

General instructions

You exam will be solely on computer. You will be given a problem set in the following and you will need to design the program, implement it, compile it, debug it and test it on computer. Before the time is up, you should **upload your solution in Sakai**. You need to name the file using the format bellow and submit it to the Sakai: "sYourID_midterm1.zip" which includes all your m-files. Note that description file is not required.

Unless otherwise indicated as part of the instructions for a specific problem, comments will not be required on the exam. Uncommented code that gets the job done will be sufficient for full credit on the problem. On the other hand, **comments may help you to get partial credit if they help us determine what you were trying to do**.

The examination is open-book, and you may make use of any texts, handouts, course notes, or any internet resources. However, you are not allowed to collaborate during the exam. Please close all communication software in your computer such as, QQ, gtalk, or any other webchat tools. And you cannot use your cell phone. **Any communication about the exam questions between you and other students is strictly prohibited**.

Exam will last 100 minutes, and have 100 points totally.

Question 1.

(25 points) Write a Matlab script to keep asking for user input of two integers a and b, then display the greatest common divisor. When input 'good luck' , exit the program. You CANNOT use Matlab functions *gcd* or *lcm*. An example run of the program is shown below:

```
Please input the first number, ("good luck" to exit): 30
Please input the second number, ("good luck" to exit): 20
The greatest common divisor = 10
Please input the first number, ("good luck" to exit): 100
Please input the second number, ("good luck" to exit): 100
The greatest common divisor = 100
Please input the first number, ("good luck" to exit): 2
Please input the second number, ("good luck" to exit): 7
The greatest common divisor = 1
Please input the first number, ("good luck" to exit):
```

Question 2.

(25 points)$a \equiv b \pmod m$ means a=mk+b, where k is a natural number. Suppose we have equations:

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad \vdots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

Let $M = m_1 \times m_2 \times \cdots \times m_n = \prod_{i=1}^{n} m_i$, $M_i = M/m_i$,and t_i is a number satisfying the equation $t_i M_i \equiv 1 \pmod{m_i}$, then $x = \sum_{i=1}^{n} a_i t_i M_i$. This algorithm is the famous Chinese remainder theorem.

Write a script to ask for user input the coefficients of equations, and if user input "begin", your script will solve the equations of the given coefficients and display the answer. The number of equations will be less than 6. An example run of the program is shown below:

```
input a1: 2
input m1: 3
input a2: 3
input m2: 5
input a3: 2
input m3: 7
input a4: GG
The solution is 23
```

Question 3.

(25 points) Given a continuous function f=x^3-19*x-30 over an interval [a; b] such that sign(f(a)) 6= sign(f(b)) find a<=r<=b such that f(r) = 0. The bisection method consists in dividing the interval [a; b] into two sub-intervals [a; c] and [c; b] of equal size. Then either f(a) and f(c) or f(c) and f(b) will have different signs. In case c = r we stop and return c, otherwise the process is repeated over the interval where the sign changes. The process of narrowing down the interval will only end when the error is smaller than a bound specified by the user. Now use this algorithm to find two solutions of f=x^3-19*x-30 over an interval [2,10] and [-10,-2.5].

Question 4.

(20 points) Write a Matlab script to solve for x in the equation ax^2 + bx +c = 0. Here we assume that a, b, c, and x are all real numbers. User will input the coefficient a, b, c, then you need to display the corresponding solution.

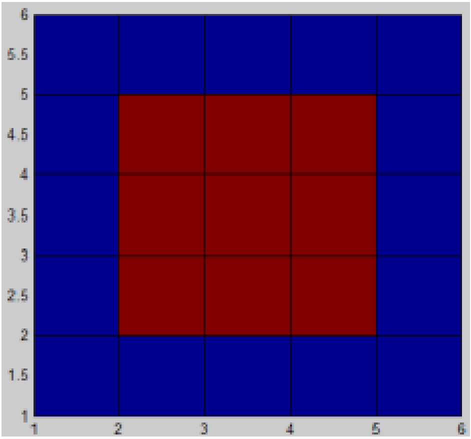
Question 5.

(a) (25 points) Write a Matlab script to read in the provided image files "1.jpg", "2.jpg", and "3.jpg", then roughly count the number of pixels that the number "1", "2", and "3" occupied in the image. Display your results. Notice that sizes of all images are the same, 65 x 95 pixels. Hint: set a threshold to separate the number area.

(b) (10 points) Write a Matlab script to recognize the number in "Recognize.jpg", you should be able to output the number "213" according to the image. Notice that the size of "Recognize.jpg" is 195 x 95 pixels and can be separated into 3 regions with size 65 x 95 pixels where each region contains one number. Hint: you can use the results in (a) to tell the difference of "1", "2", and "3".

Question 6.

(20 points) Your task is to draw a square like the graph below:



Notice that width is 5 and width of edge is 1 in the left graph. What is required is the color of the edge must be different from the color inside. Don't worry about the actual color. It's fine if you have yellow for the edge and green inside.

Please start like this:

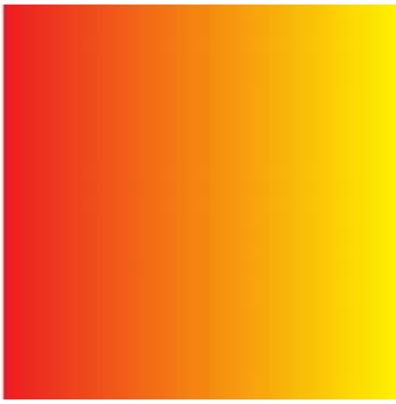
```
function draw(width, width_of_edge)
```

Hint: the color is determined by the value in the matrix.

Question 7.

(20 points)Two image files are given in the attachment as red.png and yellow.png. Your task is to use these two files to generate an image file like the following, which contains

transient color between red and yellow.



Notice that your program should also be able to process other colors such as blue and green.

Question 8.

(15 points) Write a Matlab script to read in all the prices in the provided text file “Prices.txt” and put them into an array arr1, then find the mean, the variance and the highest price. Calculate the differences between adjacent two days and save in another array arr2, where arr2(i)=arr1(i+1)-arr(i) for index I, then create a new text file “Analysis.txt” and save the mean,

the variance and the array arr2 into it. In addition, plot arr1 and arr2 against their indices in the same figure, and the line of arr2 should be in red. An example of "Analysis.txt" is shown below:

```
The mean of prices is 23.2.
The variance of prices is 0.7408.
The highest price is 24.5.
0.4
1.6
-0.5
1.4
-0.9
-0.2
-0.5
0.5
1.2
-1.1
```

Question 9.

(10 points)Now you are to implement a program that is able to deal with the following input:

"add 2 6"	2+6
"mul 5 7"	5*7
"factr 5"	5!
"3 sqrt 5"	$\sqrt[3]{5}$

Your program should have proper output to display the result.

Question 10.

(15 points)Write a program to accept user input of a string, then increment all the numbers within the string by 1, and change the lowercase letter to uppercase letter and vice versa. If the number is 9, then you should change it to 0. The following is an example run of the program:

Input: i will GET 099 poinTs in the Midterm Exam.

Output: I WILL get 100 POINtS IN THE mIDTERM eXAM.

Question 11.

(35 points) Write a program to ask the user to input a number N(less than 10), then accept user input a sentence of N words, then sort the words in dictionary order and print the result on the screen with a space between the sorted words. Here we assume that all the words contain only lowercase letters. The following is an example run:

Please input N: 5

Input: after all i don not exist for anyone

The sorted result: after all anyone do exist for i not

Cite: Novecento. Un monologo(The Legend Of 1900)

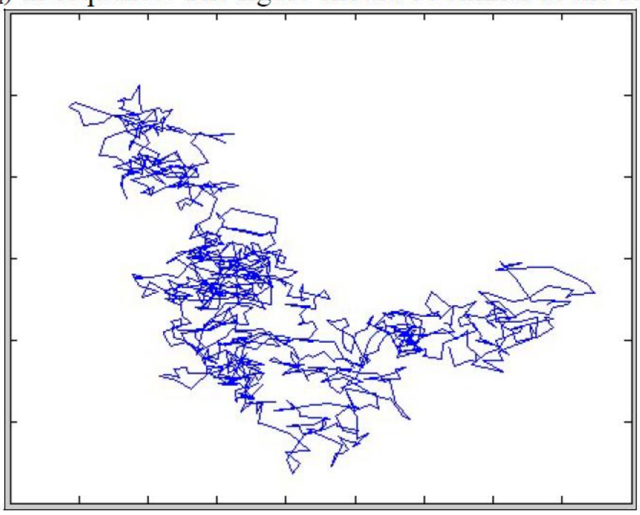
Question 12.

(35 points) Two-dimensional Brownian motion simulation

Consider the two-dimensional Brownian motion of a particle starting from the origin (0, 0). At time t = 0, the particle is at the origin (0, 0). Suppose at time t = n (n = 0, 1, 2, …) , the particle is at (x_n, y_n), then at time t = n+1, the location of the particle can be simulated as (x_{n+1}, y_{n+1}) = (x_n+ Δx, y_n + Δy), where Δx and Δy are random numbers with normal distribution (mean = 0 and standard deviation = 1). In this way, we can simulate the location of the particle at time t = n, given that the particle is starting from the origin (0,0) at t=0.

Your task is to simulation the Brownian motion of 1000 particles, all starting from the origin at t=0. In the command window, given a user input of a time n and a distance r (r is defined as sqrt(x^2+y^2)), find out the number of particles whose distance from the origin is greater than r. Display your result in the command window. At the same time,

plot the Brownian motion of the last particle in a figure by connecting all the coordinates (x_n, y_n) in sequence. The figure should be similar as the following:



Hint: use the built-in function *randn* to get the random step.

Question 13.

In this problem, you're going to use the method of RECURSION to do everything

1. (5 points) Given an array, for example A = [1 2 3 4 5], calculate the product of the first n elements, where n is provided by user.
2. (5 points) Given an array, for example A = [1 2 3 4 5], delete the last n elements and make a new array, where n is provided by user.
3. (15 points)Given an array, for example A = [1 2 3 4 5], delete the odd elements.

Question 14.

Type the code “why” again and again in your command window and try to understand the story.

Have fun and …

Good luck!

Charles Liu