CS 158/159 Lab 10

In this lab session you will complete the following tasks in the specified order. **Do not begin this lab until your lab sessions starts**. Individuals failing to remain on task during the lab will be assigned a zero for the lab, lab quiz, and will be considered absent for the lab. Web browsing, personal e-mail, social networking, and/or text messaging is not permitted during lab.

1. Record Attendance

Log on to your UNIX account on guru on a physical PC in the lab room (not on a wireless device) and enter the command **`attend`** to officially record your lab attendance. A confirmation e-mail will then be sent to you to verify that your attendance was correctly recorded. If you are late, you should still run the `attend` command and continue to participate in the remaining portion of the lab, but you will not be eligible to receive credit for the lab programming assignment. If you are present in the lab, but fail to run the `attend` command as expected, you will be marked as absent and likewise will not be eligible to receive credit for the lab programming assignment.

2. Review Material

The first task is to improve your understanding of the material from the book and the course programming standards (available in the course packet and in Blackboard) by working through a series of problems with the assigned lab partners of your group. You must show your lab instructor that you have successfully completed these tasks BEFORE you leave lab today. You will not receive points for the programming assignment, unless this task has been completed to the satisfaction of your lab instructor.

Redirection Problems: Given an executable file (`a.out`) that accepts input from the keyboard and produces output to the monitor.

How would you redirect input from a file called `dataset` into the executable file?

How would you redirect both input to the executable file from a file called `dataset` and redirect output to another text file called `results`?

What takes place when you attempt to redirect output using > to an existing file?

How would you redirect output to append to the end of an existing file called results?

Solve the following problems related to material from external I/O in Octave/MATLAB:

Statement	**True / False**
The parameter to the `feof` function is a file handle.	
The value returned from the `fopen` function is used to initialize a file handle.	
The name of the external file is the parameter to the `fclose` function.	
In Octave/MATLAB the `fclose` function can only be used to a close a connection that has been successfully opened.	
The `fscanf` function references the name of the file handle but NOT the name of the external file (as one of its parameters).	
When the file handle stores a file ID value equal to zero it means the external file failed to open as expected.	
A file handle used to read external data into a program should always be compared with an error value to determine the data file was opened as expected before allowing the program to continue.	
The value returned from the `fscanf` function is the data value that was read from the external file.	
The `feof` function will return a zero when the end of the file has been reached.	
The name of the external file being accessed is case insensitive when referenced in the `fopen` function call.	
The file extension (`.dat`, `.txt`) in the name of the external file being accessed does not have to be referenced in the `fopen` function call.	
A loop that uses the result of the `feof` function as a control expression will iterate the same number of times as there are items in the data file to read (assume one scalar value per iteration is read).	

3. Programming Assignment

The second task is to develop a program as a group which solves the given problem. This assignment is worth 15 points and will be **due 30 minutes prior to the start of your next lab session**. All assignment deadlines are firm and the ability to submit your assignment will be disabled after your deadline elapses. No late work will be accepted!

As you develop the program, you should rotate through the following roles approximately every 30 minutes. Do not allow the same person be the driver the entire time. It is not acceptable to designate a single individual to complete the assignment. Every individual group member should have a full understanding of all work submitted. Assignments are an opportunity to develop and demonstrate your understanding of course material.

Role	**Description**
Driver	The driver is in charge of the computer which includes entering code, saving, testing, and submitting. This individual should be soliciting the other two members for advice.
Navigator	The navigator's role is to look over the shoulder of the driver for syntactical errors, logical errors, and concerns related to course standards. The most common mistakes include failing to pair quotes, misplaced semi-colons, and improper placement of parentheses.
Manager	The manager may not be as close to the driver as the navigator but still plays an important role ensuring that the algorithm to be implemented is correct and can be tested using a variety of input to verify correctness. The manager is responsible for communicating to the teaching assistant who will be making the group's final lab submission.

This programming assignment does not have a single solution, and each group should collaborate together to develop their own unique solution. Submissions may be processed with comparison software and results will be used to detect unacceptable collaboration between groups. The development of your algorithm and the resulting code should only be discussed among your group members and course staff.

Your program must adhere to the course programming standards (available in the course packet and in Blackboard). Please review this document before submitting your assignment, because failure to adhere to it will result in a reduction in points. You program must include the designated program header (~cs15x/student/hdrProg) at the top of the program (which can be inserted in vi using the hp shortcut while in command mode). **The header must be converted to use the proper MATLAB comment characters**. The header must include an appropriate description of your program and must contain the official Purdue career account e-mail addresses of each **contributing** group member. Do not include the e-mail addresses of people who did not participate in the program development. Failing to participate in the process to the satisfaction of all partners will result in a zero. Also note that course standards prohibit the use of advanced programming concepts not yet introduced in the course, unless otherwise specified. Each of the example executions provided below represents a single execution of the program. Your program must accept input and produce output **exactly** as demonstrated in the example executions. Your program will be tested with the data seen in the examples below and an unknown number of additional tests making use of reasonable data. Do not include any example outputs with your submission.

A single program file (**with the .m extension**) will be submitted electronically via the guru server. An example submission was conducted during the first week in lab00. If you have a concern regarding how to submit work, please visit course staff prior to the assignment deadline.

Problem: A given data file will contain an unknown number of integers. The first integer represents the target value and the remaining values will be tested to determine whether they can reach the target value through zero or more integer divisions by the number two. Display, to the monitor, those integers present that can be reduced to the target value and the total count.

This program is to be written in Octave and all data should be read from a file called `testdata`.

Example Execution #1: (`testdata: 2 11 2 24 17 9 21`)
```
Values that reduce to the target [2]: 11 2 17 9 21
Total number of values reduced to the target: 5
```

Example Execution #2: (`testdata: 3 11 2 24 17 9 21`)
```
Values that reduce to the target [3]: 24
Total number of values reduced to the target: 1
```

Example Execution #3: (`testdata: 4 11 2 24 17 9 21`)
```
Values that reduce to the target [4]: 17 9
Total number of values reduced to the target: 2
```

Example Execution #4: (`testdata: 13 61 73 59 47 31 91 101 29`)
```
None of the present data can be reduced to the target.
```

Example Execution #5: (`testdata: 8 984 1574 1009 329 589 1589 69 425 1732 1025 963 145 1174 1284 1475 805 202 438 138 758 1889 886 1709 1247 1436 388 299 1813 965 571 517 1253 127 1526 1583 21 401 956 446 115 1982 1409 260 1138 1998 1039 1247 182 1478 689`)
```
Values that reduce to the target [8]: 69 1025 138 571 517 260 1138 1039
Total number of values reduced to the target: 8
```

Additional Notes:
- This program is to be written in Octave (a program largely compatible with MATLAB) and submitted on `guru` with a `.m` file extension.
- You will continue to use the normal format of the block comment headers (including the Purdue e-mail addresses of the group members), but you will have to change the comment characters to % or %{ and %}.
- Your program MUST open an external file named exactly `testdata` (do not add an extension to or use any variation of this file name). Referencing any other external file will result in your program failing all test cases and will be graded accordingly.
- Your program must accept input and produce output in the same manner as seen in the examples provided. All output is to be displayed on the monitor and the contents of `testdata` are not to be displayed with the output.
- This is an end of file problem as you do not know the quantity of data that will be in the external file. Only non-negative integers and white spaces will be found in the data file.
- Standard functions that are permissible for use on this assignment: `fopen`, `feof`, `fclose`, `fscanf`, `fprintf`, `mod`, `floor`, `abs`. The use of any other standard function must be approved by your lab instructor. **No user-defined functions are to be used**. All variables in your program must be scalar, no arrays, vectors, matrices are permitted on this assignment.
- Course standards prohibit the use of programming concepts not yet introduced in lecture. For this assignment, you can only consider material in the first 7 chapters of the book, notes, and lectures. Use of advanced programming constructs beyond this material would result in a loss of points.

4. Group Coordination

The next task is to collaborate as a group to determine who will submit your program assignment for grading. **Only one person per group will make submissions for the entire group**. Lab instructors are not required to grade submissions from multiple members of the same group to determine which submission you actually want graded. Also, the group member should not always be making the submission each week. Record the names and official Purdue career account e-mail addresses of all three lab partners here and put a checkmark in the Submitter column for the person responsible for making the submission. Your group should turn in this page of information to their lab instructor before you leave lab today.

Name	Purdue career account e-mail address	Submitter

If possible, it is a good idea to submit your work for grading prior to leaving lab today, even if it is not completely finished. This will allow each lab partner to **verify their contact information in the assignment header is correct** as each would then receive an e-mail verifying the submission. You may make multiple submissions before the deadline, but only the last attempt is retained and graded. Any previous submissions will be over-written and cannot be recovered. The submission script will reject the submission of any file that does not compile. A program must compile to be considered for partial credit.

If your group is unable to complete your assignment during the lab session, then it is expected that your group will meet outside of class to finish and submit the programming assignment. That is why you should exchange contact information during lab today! Before you leave, you should discuss when and where the group will meet next and when you plan for final submission to be made. If a group member you have entrusted to do the submission cannot be contacted and he/she is the only one who has a copy of the program, then the rest of the group would have to essentially start over in order to complete the programming assignment. Thus, it is a good idea for each person to have a copy of the program.

Consider the members of your group to be resources to assist you as you learn the material in this course. As a group, you may want to arrange a time to visit the TA office hours together or attend a particular Supplemental Instruction session together.

5. Lab Quiz

The final task will be to take the lab quiz which be made available in Blackboard during the last part of your lab today. **Lab quizzes are individual efforts and you may not use any resources while completing the quiz.** The quiz is worth 5 points and you must be present in lab session in order to credit for the quiz. All problems on lab quizzes will be multiple-choice or true-false. The quiz will emphasize material from book and the appropriate course programming standards. You will be given 10 minutes to complete the quiz and questions will be presented one at a time and cannot be revisited. Be sure to save your answers to each question and to finish your quiz to ensure it is submitted for grading. Any quiz that is taken when it is not being proctored by the lab instructor will receive zero points.