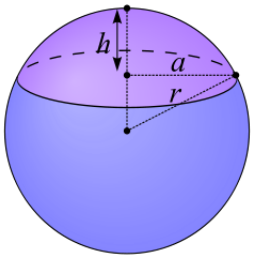
CS1132 Fall 2013 Assignment 1a due 9/10 11:59pm

Adhere to the Code of Academic Integrity. You may discuss background issues and general strategies with others and seek help from course staff, but the implementations that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is never OK for you to see or hear another student's code and it is never OK to copy code from published/Internet sources. If you feel that you cannot complete the assignment on you own, seek help from the course staff.

When submitting your assignment, follow the instructions summarized in Section 4 of this document.

Do not use the `break` or `continue` statement in any homework or test in CS1132.

1 Spherical Cap



The area and volume of a spherical cap are given by the following equations [1]

$$A = 2\pi r h$$
$$V = \frac{\pi h^2}{3}(3r - h)$$

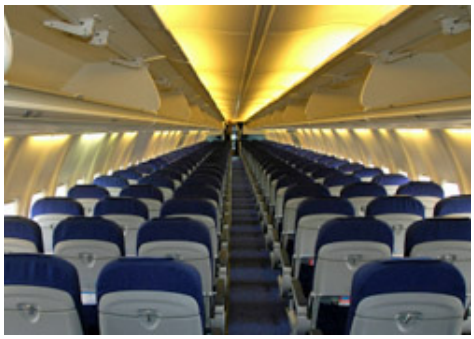
where r stands for the radius of the sphere, and h is the height of the cap. The constraints on h and r are $h > 0$, $r > 0$ and $h < 2r$. Therefore, if input argument is out of range, the output should be 0 for A and V. The file *SphericalCap.m* defines a function which takes r and h as input arguments, and returns A the area and V the volume as output.

- Implement the following function:

```
function [A, V] = SphericalCap(r,h)
% This function computes the area and the volume of a spherical cap
% Input (r,h):  r: the radius of the sphere; h: the height of the cap.
% Output (A,V): A: the area; V: the volume
```

[1] Image is from Wikipedia

2 Airplane Seating Assignment



Write a program that can be used to assign seats for a commercial airplane. The airplane has 10 rows, with 4 seats in each row. Rows 1 to 2 are first class, rows 3 through 5 are business class and rows 6 to 10 are economy class.

(i) Random integer generator

Using the `help` command, learn about the `rand, ceil, fix`, and `floor` functions. After reading their brief description, implement a random integer generator using those functions only. Implement the following function:

```
function result = myRandInt(startInt,endInt)
% Returns an integer from startInt to endInt, inclusive with equal probability.
% If startInt > endInt, treat startInt as the upper bound and endInt as the lower
% bound instead.
```

(ii) Random seat assignment

We would like to randomly select a seat for a given passenger. The random assignment is achieved by selecting the row number from 1, 2, $\cdots$, or 10 with equal probability. Therefore, the probability for choosing each row is the same. Then the class type is a result of the random assignment of row number. So the likelihood for getting each class is different. Within each row, each seat is still chosen with equal likelihood. Make effective use of function `myRandInt` from the previous problem. Implement the following function:

```
function [rClass,rRow,rSeat] = RandomChooseSeat()
% Pick a seat at random from the airplane which contains 10 rows
% rClass represents the choice of class
% 1 represents first, 2 represents business and 3 represents economy.
% rRow represents the row number from 1, 2, ..., or 10.
% rSeat represents the seat number from 1, 2, 3, or  4.
```

(iii) Random unique seat assignment

When randomly choosing a seat, we should know which seats are reserved. Therefore, we use a binary row vector of length 40 called `seatChart` to represent the plane's seats availability, where $seatChart(1)$ is seat 1 in row 1, $seatChart(5)$ is seat 1 in row 2, and so forth. If a seat has been taken, then its corresponding value is 0. Otherwise the corresponding value is 1. Make effective use of function `RandomChooseSeat` from the previous problem.
Implement the following function:

```
function [rClass,rRow,rSeat,seatChart] = RandomChooseUniqueSeat(seatChart)
% Select a unique seat at random from the air plane which contains 10 rows 4 columns.
% input: seatChart is a row vector 1x40 which contains the availability of each seat.
% 1 means available, 0 means occupied.
% rClass represents the choice of class: 1 represents first, 2 represents business and
% 3 represents economy.
% rRow represents the row number from 1, 2, ..., or 10.
% rSeat represents the seat number from 1, 2, 3, or  4.
% seatChart returns a new seat chart with the updated value
```

(iv) Seating print out

Now you have a list of customers waiting for their seats to be assigned. Your plane is currently empty with no seats reserved. Those customers do not have any preferences. Their names are stored in a mat file called `nameList.mat`. In order to access `namelist` which stores the list of names, use the command `load nameList.mat`. In addition, use the expression `nameList(n,:)` to access the nth person's name.
Write a script `PrintSeat.m` to display the first 10 customers' random unique seats chosen from your plane. Make effective use of function `RandomChooseUniqueSeat` from the previous problem.
Display each seat by first printing out the customer name and then its seat class type, row number, and seat number.

```
Name      Class#      Row#    Seat#
Lianne     first        1       2
Lisette    economy      8       1
Lita       business     3       1
```

(v) Seat assignment for groups

Now you have an airplane with some seats occupied. A new customer wants to book tickets for a group with N ($N \leq 4$) members and prefers those N people to sit as close to one another as possible. They really love their group sitting in the same row consecutive to each other. However, if that is not possible, they would like to sit in the same row. If none of the above can be satisfied, they will simply not booking the tickets.
Implement the following function:

```
function [rClass,rRow,rSeat,seatChart] = ReserveForGroups(seatChart,N)
% Select n number of seats from the airplane which contains 10 rows 4 columns.
% input: seatChart is a row vector 1x40 which contains the availability of each seat.
% 1 means open, 0 means occupied.
% N indicates the size of group N > 0 && N<=4
% rClass represents the vector for the choice of the group
% rRow represents the vector for the row number from 1, 2, ..., or 10.
% rSeat represents the seat number from 1, 2, 3, or  4.
% seatChart returns a new seat chart with the updated value
% if the group cannot be booked, then rClass, rRow, rSeat will be []
```

3 Self-check list

The following is a list of the minimum *necessary* criteria that your assignment must meet in order to be considered *satisfactory*. Failure to satisfy any of these conditions will result in an immediate request to resubmit your assignment. Save yourself and the graders time and effort by going over it before submitting your assignment for the first time. Although all these criteria are necessary, meeting all of them might still not be *sufficient* to consider your submission satisfactory. We cannot list everything that could be possibly wrong with any particular assignment!

- Δ Comment your code! Make sure your functions are properly commented, regarding function purpose and input/output parameters.
- Δ Suppress all unnecessary output by placing semicolons (;) appropriately. At the same time, make sure that all output that your program intentionally produces is formatted in a user-friendly way.
- Δ Make sure your functions names are *exactly* the ones we have specified, *including* case.
- Δ Check that the number and order of input and output parameters for each of the functions match exactly the specifications we have given.
- Δ Test each one of your functions independently, whenever possible, or write short scripts to test them.
- Δ Check that your scripts do not crash (i.e., end unexpectedly with an error message) or run into infinite loops. Check your script several times in a row. Before each test run, type the commands `clear all; close all;` to delete all variables in the workspace and close all figure windows.

4 Submission instructions

1. Upload files `SphericalCap.m`, `myRandInt.m`, `RandomChooseSeat.m`, `RandomChooseUniqueSeat.m`, `PrintSeat.m`, and `ReserveForGroups.m` to CMS in the submission area corresponding to Assignment 1a in CMS before the deadline. Late submission is accepted up to 24 hours after the deadline with a 10% penalty.
2. *After grading:* If you resubmit the assignment, upload your corrected files and *be sure to select the* ***Regrade Request*** *option.*