EE 289 – Homework Chapter 4

1 Write a script to solve this problem. Assume that you have a vector named D. Using iteration (for and /or while) and conditionals (if and/or switch), separate vector D into four vectors posEven, negEven, posOdd, and negOdd.

- posEven contains all of the positive even numbers in D.
- negEven contains all of the negative even numbers in D.
- posOdd contains all of the positive odd numbers in D.
- negOdd contains all of the negative odd numbers in D.

For example:
if D=[-4 -3 -2 -1 0 1 2 3 4],
posEven=[2 4], negEven=[-4 -2], posOdd=[1 3]; negOdd=[-3 -1]

```
clear
clc
D = [-4,-3,-2,-1,0,1,2,3,4]
% posEven=[2,4], negEven=[-4,-2],
% posOdd=[1,3] and negOdd=[-3,-1]
posEven = []; posOdd = []; negEven = []; negOdd = [];
for val = D
    if val > 0
        if mod(val,2) == 0
            posEven = [posEven val]
        else
            posOdd = [posOdd val]
        end
    elseif val < 0
        if mod(val,2) == 0
            negEven = [negEven val]
        else
            negOdd = [negOdd val]
        end
    end
end
```

3 You are hiring grad students to work for your company, which you have recently started. The HR department has asked you to write a script that will help them determine the chances of an individual applicant getting a job after interviewing. The following table outlines the for determining the chances for the applicant to get a job:

GPA value	Chance of Being Hired
GPA >= 3.5	90%
3.0 <= GPA < 3.5	80%
2.5 <= GPA < 3.0	70%
2.0 <= GPA < 2.5	60%
1.5 <= GPA < 2.0	40%
GPA < 1.5	30%

Your script should repeatedly ask the user for a GPA value and compute the student's chances of being hired. It should continue asking for GPA values until a negative number is entered. For example:

- GPA input: 4 should give the answer 0.9
- GPA input: 3.4 should give the answer 0.8

```
clear
clc
% GPA input: 4 should give the answer 0.9
% GPA input: 3.5 should give the answer 0.9
% GPA input: 3.4 should give the answer 0.8
GPA = 1;
while GPA > 0
    GPA = input('GPA: ');
    if GPA > 0
            if GPA >= 3.5, answer = 0.9
        elseif GPA >= 3,   answer = 0.8
        elseif GPA >= 2.5, answer = 0.7
        elseif GPA >= 2,   answer = 0.6
        elseif GPA >= 1.5, answer = 0.4
            else           answer = 0.3
        end
    end
end
```

5 You have a friend who has too many clothes to store in his or her tiny wardrobe. Being a good friend, you offer to help to decide whether each piece of clothing is worth saving. You decide to write a script that will compute the value of each piece of clothing. A piece of clothing has give attributes that can be used to determine its value. The attributes are: condition, color, price, number of matches, and comfort. Each attribute will be rated on a scale of 1 to 5. Write a script called clothes that will ask the user for the ratings for each attribute and store the result in a vector. The order of attributes in the vector is: [condition color price matches comfort]

The script should compute a value between 0 and 100; 100 represents a good piece of clothing, while 0 represents a bad piece of clothing. The points that should be given for each attribute are shown below:

Condition:	1=>0; 2=>5; 3=>10; 4=>15; 5=>20

Color:	1=>blue=>12;
	2=>red=>2;
	3=>pink=>15;
	4=>yellow=>20;
	5=>white=>12

Price:	1=>8,2-3=>16,4-5=>20
Matches:	1-2=>8,3-5=>19
Comfort:	1=>6,2-3=>13,4-5=>18

Note: If a number other than 1-5 is assigned for one of the attributes, no points should be given.

```
clear
clc
quit = false;
values = [0 5 10 15 20;
          12 2 15 20 12;
          8 16 16 20 20;
          8 8 19 19 19;
          6 13 13 18 18]
while ~quit
    cond = input('condition: ','s');
    quit = cond < '1' || cond > '5';
    if ~quit
        vec(1) = cond - '0';
        vec(2) = input('color: ');
        vec(3) = input('price: ');
        vec(4) = input('matches: ');
        vec(5) = input('comfort: ');
        value = 0;
        for ndx = 1:5
            if vec(ndx) > 0 && vec(ndx) < 6
                value = value + values(ndx, vec(ndx));
            end
        end
        fprintf('value = %d\n', value)
    end
end
```

7 Now that you're comfortable with iteration, you're going to have to solve an interesting problem. It seems that the Math department at a rival university has once again dropped the ball, and forgotten the value of pi. You are to write a function called mypi, which consumes a number that specifies the required accuracy and then approximated the value of pi to that accuracy. You are going to use the following algorithm based on geometric probability.

Think about a quarter circle inside a unit square (the quarter circle has area $\pi/4$). You pick a random point inside the square. If it is in the quarter circle, you get a "hit"; and if not, you get a "miss." The approximate area of the quarter circle will be given by the number of hits divided by the number of points you chose.

Your function should repeat the process of counting hits and misses until at least 10,000 tries have been made, and the successive estimates of pi are within the prescribed accuracy. If should return the estimate value of pi.

```
clear
clc
N = 10;
for power = 1:7
    hits = 0;
    misses = 0;
    for iter = 1:N
        pt = rand(1,2);
        dist = sum(pt.^2);
        if dist < 1
            hits = hits + 1;
        else
            misses = misses + 1;
        end
    end
    fprintf('estimate of pi is %5.4f\n', 4 .* hits ./ N)
    N = N .* 10;
end
```