CS100 – Fall 2015

Homework #2

Due: December 4th , 2015, 23:55

Problems

1) Write a Matlab/Octave function named **convertMillis** that converts milliseconds to hours, minutes and seconds and milliseconds. Your function will have the following header:

```
function [hours, minutes, seconds, millis] = convertMillis(millis)
```

The method returns an array of integers as `[hours, minutes, seconds, millis]`. Example runs for the function are as follows:

```
>>[h m s ms]=convertMillis(5500)
h = 0
m = 0
s =  5
ms =  500

>>[h m s ms]=convertMillis(550000)
h = 0
m =  9
s =  10
ms = 0
```

```
>>[h m s ms]=convertMillis(550320)
h = 0
m =  9
s =  10
ms =  320

>>[h m s ms]=convertMillis(12550320)
h =  3
m =  29
s =  10
ms =  320
```

2) A regular polygon is an n-sided polygon in which all sides are of the same length (*s*) and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). The formula for computing the area of a regular polygon is

$$Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\Pi}{n}\right)}$$

Write a **function** named **calculateArea** with the following header that receives the number of sides and the length of a side of a regular polygon and returns its area by using the above formula.

```
function area = calculateArea(n,s)
```

Example runs for the function are as follows:

```
>> calculateArea(5,5)
ans =  43.012
```

```
>> calculateArea(12,5)
ans =  279.90
```

3) (Math: approximate the square root) There are several techniques for implementing the sqrt method in the Matlab/Octave library. On such technique is known as the Babylonian method. It approximates the square root of a number, **n**, by repeatedly performing a calculation using the following formula:

nextGuess = (**lastGuess** + **n** / **lastGuess**) / 2

When **nextGuess** and **lastGuess** are almost identical, **nextGuess** is the approximated square root. The initial guess can be any positive value (e.g., 1). This value will be the starting value for **lastGuess**. If the difference between **nextGuess** and **lastGuess** is less than a given epsilon (a small number), you can claim that **nextGuess** is the approximated square root of **n**. If not, **nextGuess** becomes **lastGuess** and the approximation process continues. Implement the following method that returns the square root of **n**.

```
function result= mysqrt(n,e)
```

Example runs for the function are as follows:

```
>> mysqrt(4,0.1)
ans =   2.0006

>> mysqrt(4,0.01)
ans =   2.0000
```

```
>> mysqrt(9,0.1)
ans =   3.0001

>> mysqrt(9,0.01)
ans =   3.0000
```

Note:

- When submitting, submit three files named `convertMillis`.m , `calculateArea`.m and `mysqrt`.m.
- No other methods accepted. You may resubmit as many times as you want until the deadline.
- Write your name, id and department name at the top line of each submited file in a commented manner. Ex: % Özgür Yurtsever, S011919, Industrial Eng.
- WARNING: This homework is an individual assignment. Your programs are checked and compared against each other using automated tools. Any act of cheating will be punished. DO NOT GIVE/TAKE YOUR HOMEWORK TO/FROM OTHERS.