Tutorial Task 1

Write three MATLAB/OCTAVE functions that can call each other, each function takes an input, the number or the name of the fuction called it, and it is called, the function displays a message that it was called by such and such.

Example

Fun1, func2, func3

In func1 if you call func3(1) it will display
I was invoked by func1

A solution

```
%%
```

Tutorial Task 2

The distance a freely falling object travels is

$$x = \frac{1}{2} g t^2$$

where

g = acceleration due to gravity, 9.8 $\mathrm{m/s^2}$
t = time in seconds
x = distance traveled in meters.

If you have taken calculus, you know that we can find the velocity of the object by taking the derivative of the preceding equation. That is,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v = gt$$

We can find the acceleration by taking the derivative again:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = a = g$$

(a) Create a function called `free_fall` with a single input vector `t` that returns values for distance `x`, velocity `v`, and acceleration `g`.
(b) Test your function with a time vector that ranges from 0 to 20 seconds.

Tutorial Task 3

Create a function called polygon that draws a polygon with any number of sides. Your function should require a single input: the number of sides desired. It should not return any value to the command window but should draw the requested polygon in polar coordinates.

A solution

Tutorial Task 4

The future-value-of-money formula relates how much a current investment will be worth in the future, assuming a constant interest rate:

$$FV = PV \times (1 + I)^n$$

where

FV is the future value
PV is the present value or investment
I is the interest rate expressed as a fractional amount per compounding period—i.e., 5% is expressed as .05
N is the number of compounding periods.

(a) Create a MATLAB® function called `future_value` with three inputs: the investment (present value), the interest rate expressed as a fraction, and the number of compounding periods.

(b) Use your function to determine the value of a $1000 investment in 10 years, assuming the interest rate is 0.5% per month, and the interest is compounded monthly.

A solution

HW Task 1

You are playing a game where you roll a die 10 times.
If you roll a 5 or 6 seven or more times, you win 2 dollars; four or more times, you win 1 dollar; and if you roll a 5 or 6 three or less times, you win no money.

Write a function called diceGame that takes in a vector representing the die values and returns the amount of money won.
For example:

diceGame([5 1 4 6 5 5 6 6 5 2]) should return 2
diceGame([2 4 1 3 6 6 6 4 5 3]) should return 1
diceGame([1 4 3 2 5 3 4 2 6 5]) should return 0

Note: This function should work for any length vector.

HW Task 2

Improve the previous problem by investigating the randi function (generates a random integer).
So instead of itering the result of the die manually, you call a function that runs a die 10 times, and then checks your score.

You should check how to limit randi, to give you integers between 1 and 6 only.

HW Task 3

Solve this using **for loops inside the function**, and using the build in function (transpose)

Write a function called trans(A). This particular function should take in a N by M array, A, find the transpose of A, and store in the array Atrans.

$$A\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix} \quad A^T\begin{bmatrix}1&4&7\\2&5&8\\3&6&9\end{bmatrix}$$

$$A\begin{bmatrix}1&4&3\\8&2&6\\7&8&3\\4&9&6\\7&8&1\end{bmatrix} \quad A^T\begin{bmatrix}1&8&7&4&7\\4&2&8&9&8\\3&6&3&6&1\end{bmatrix}$$