

**Date: 22-11-2021**

## **Experiment 9**

**Aim:** To work with input/output functions in MATLAB.

**Apparatus:** MATLAB Software

**Objective:** To learn about MATLAB's input/output capabilities.

**Problems:**

**Q-1** The acceleration due to the Earth's gravity at any height  $h$  above the surface of the Earth is given by the equation

$$g = -G \frac{M}{(R + h)^2}$$

where  $G$  is the gravitational constant ( $6.672 \times 10^{-11} \text{ N m}^2 / \text{kg}^2$ ),  $M$  is the mass of the earth ( $5.98 \times 10^{24} \text{ kg}$ ),  $R$  is the mean radius of the Earth (6371 km), and  $h$  is the height above the Earth's surface. If  $M$  is measured in kg and  $R$  and  $h$  in meters, then the resulting acceleration will be in units of meters per second squared. Write a program to calculate the acceleration due to the Earth's gravity in 500 km increments at heights from 0 km to 40,000 km above the surface of the Earth. Print out the results in a table of height versus acceleration with appropriate labels, including the units of the output values. Plot the data as well.

**Code:**

```
clc;
clear all;
close all;

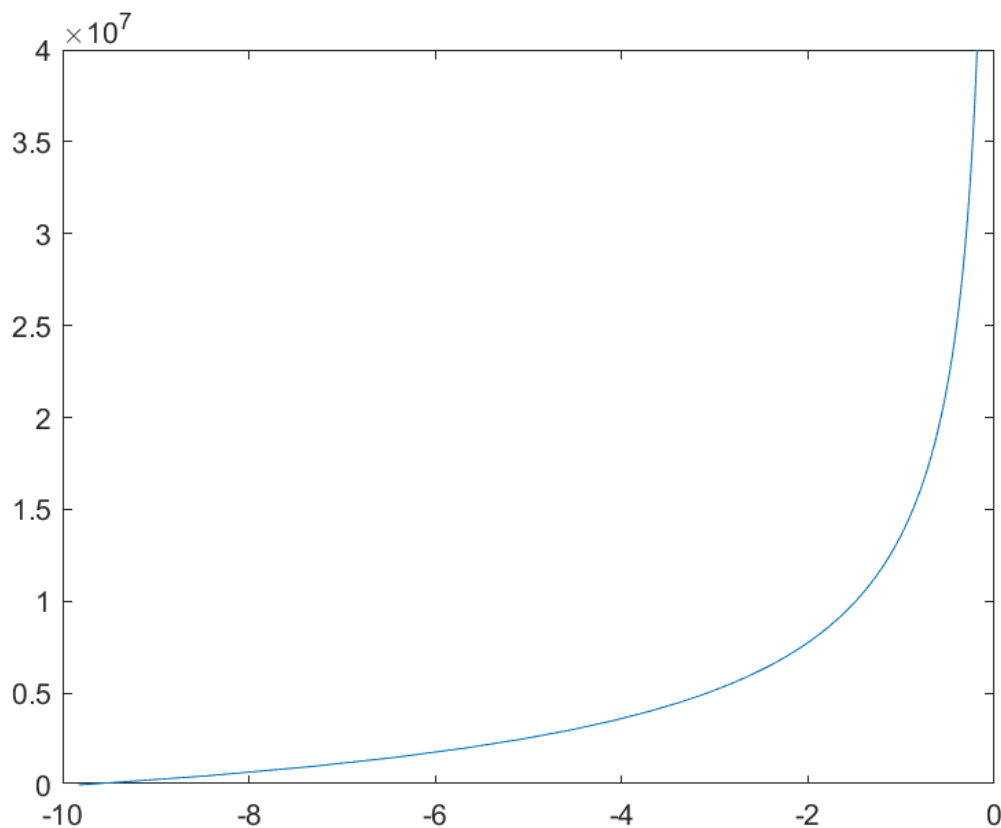
G=6.672e-11;
M=5.98e24;
R=6.371e6;
h=0:5e5:4e7;
g=-G.*M./((R+h).^2);

fprintf('Height\tGrav. Acc.\n');

for ii=1:length(g)
    fprintf('%5d\t%5d\n', h(ii), g(ii));
end

plot(g,h)
```

## Output:



**Q-2.** Write a program that reads an arbitrary number of real values from a user specified input data file, rounds the values to the nearest integer, and writes the integers out to a user-specified output file. Make sure that the input file exists, and if not, tell the user and ask for another input file. If the output file exists, ask the user whether or not to delete it. If not, prompt for a different output file name.

## Code:

```
clc;
clear all;
close all;

if ~isfile('InputFile.txt')
    fprintf('File do not exist');
    quit
end

fileID=fopen('InputFile.txt', 'r');
A=fscanf(fileID, '%f');
fclose(fileID);
A;
A=round(A);
```

```

if isfile('OutputFile.txt') —
    user=input("Do you want to overwrite the
file'OutputFile.txt'?[Y/N]", 's'); .
    if(user=='Y') ✓
        file=fopen('OutputFile.txt', 'wt'); —
    else
        name=input("Enter the name of the new file: ", 's'); —
        file=fopen(strcat(name, '.txt'), 'wt'); _
    end
else
    file=fopen('OutputFile.txt', 'wt'); ✓
end

for ii=1:length(A)
    fprintf(file, '%d\n', A(ii)); } ✓
end

fclose(file);

```

## Output:

Desktop > SEM 5 > MATLAB > LAB Laptop Copy > LAB 9 <span>Search LAB 9</span>				
Name	Date modified	Type	Size	
19BCE253_MT_PRAC-9.docx	22-11-2021 01:41 PM	Microsoft Word D...	19 KB	
input.txt	22-11-2021 01:44 PM	Text Document	1 KB	
InputFile.txt	22-11-2021 02:00 PM	Text Document	1 KB	
OutputFile.txt	22-11-2021 02:01 PM	Text Document	1 KB	
prac9_q2.m	22-11-2021 02:00 PM	MATLAB Code	1 KB	
prac9_q3.m	22-11-2021 01:49 PM	MATLAB Code	1 KB	
prac9_q4.m	22-11-2021 01:44 PM	MATLAB Code	1 KB	
prac9_q5.m	22-11-2021 01:40 PM	MATLAB Code	1 KB	










```
InputFile.txt - Notepad
File Edit Format View Help
MATLAB is my Favourite Subject

-Kanisha Shah
```

```
OutputFile.txt - Notepad
File Edit Format View Help
I love MATLAB

-Kanisha Shah
```

```
Command Window
Do you want to overwrite the file 'OutputFile.txt'? [Y/N] N
Enter the name of the new file: Kanisha_19BCE253
fx >>
```

Name	Date modified	Type	Size
 19BCE253_MT_PRAC-9.docx	22-11-2021 02:03 PM	Microsoft Word D...	261 KB
 input.txt	22-11-2021 01:44 PM	Text Document	1 KB
 InputFile.txt	22-11-2021 02:00 PM	Text Document	1 KB
 Kanisha_19BCE253.txt	22-11-2021 02:03 PM	Text Document	0 KB
 OutputFile.txt	22-11-2021 02:03 PM	Text Document	0 KB
 prac9_q2.m	22-11-2021 02:00 PM	MATLAB Code	1 KB
 prac9_q3.m	22-11-2021 01:49 PM	MATLAB Code	1 KB
 prac9_q4.m	22-11-2021 01:44 PM	MATLAB Code	1 KB
 prac9_q5.m	22-11-2021 01:40 PM	MATLAB Code	1 KB

**Q-3.** Write a program to generate a table containing the sine and cosine of for between  $0^\circ$  and  $90^\circ$ , in  $1^\circ$  increments. The program should properly label each of the column in the table.

**Code:**

```
clc;
clear all;
close all;

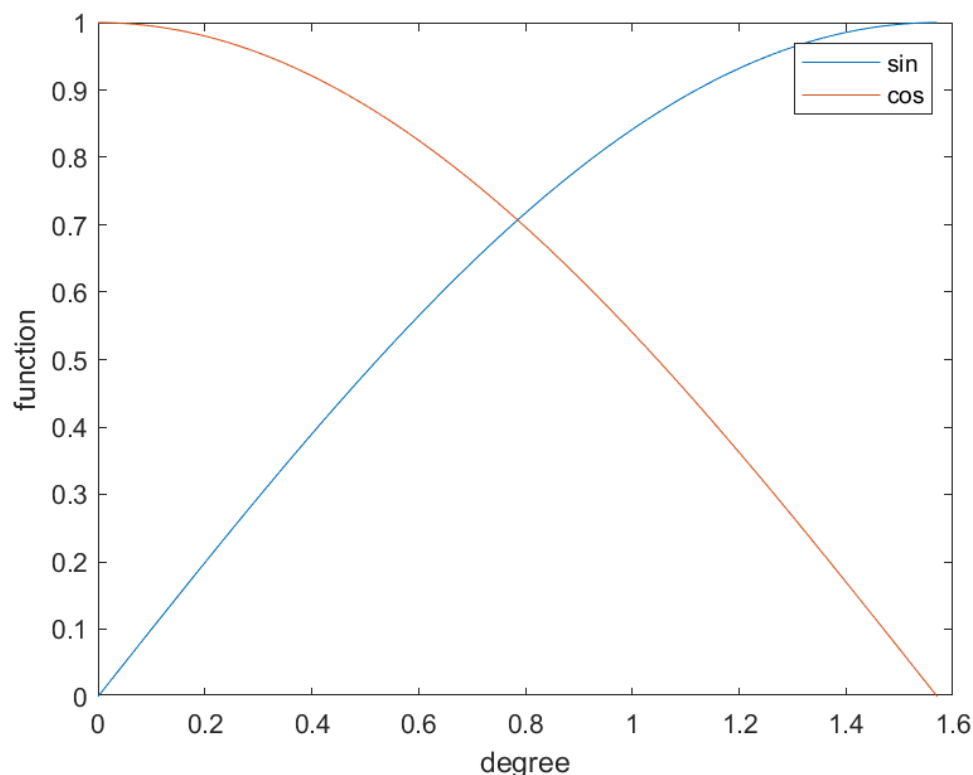
degree=0:1:90;
degree=degree.*pi./180;

for ii=1:length(degree)
    fprintf('sin(%d) = %f\t\tcos(%d) = %f\n', ii-1,sin(degree(ii)),
ii-1, cos(degree(ii)));
end

figure(1)
plot(degree, sin(degree))
hold on

plot(degree, cos(degree))
legend('sin','cos')
xlabel('degree')
ylabel('function')
```

**Output:**



Command Window

sin(0) = 0.000000	cos(0) = 1.000000
sin(1) = 0.017452	cos(1) = 0.999848
sin(2) = 0.034899	cos(2) = 0.999391
sin(3) = 0.052336	cos(3) = 0.998630
sin(4) = 0.069756	cos(4) = 0.997564
sin(5) = 0.087156	cos(5) = 0.996195
sin(6) = 0.104528	cos(6) = 0.994522
sin(7) = 0.121869	cos(7) = 0.992546
sin(8) = 0.139173	cos(8) = 0.990268
sin(9) = 0.156434	cos(9) = 0.987688
sin(10) = 0.173648	cos(10) = 0.984808
sin(11) = 0.190809	cos(11) = 0.981627
sin(12) = 0.207912	cos(12) = 0.978148
sin(13) = 0.224951	cos(13) = 0.974370
sin(14) = 0.241922	cos(14) = 0.970296
sin(15) = 0.258819	cos(15) = 0.965926
sin(16) = 0.275637	cos(16) = 0.961262
sin(17) = 0.292372	cos(17) = 0.956305
sin(18) = 0.309017	cos(18) = 0.951057
sin(19) = 0.325568	cos(19) = 0.945519
sin(20) = 0.342020	cos(20) = 0.939693
sin(21) = 0.358368	cos(21) = 0.933580
sin(22) = 0.374607	cos(22) = 0.927184
sin(23) = 0.390731	cos(23) = 0.920505
sin(24) = 0.406737	cos(24) = 0.913545
sin(25) = 0.422618	cos(25) = 0.906308
sin(26) = 0.438371	cos(26) = 0.898794
sin(27) = 0.453990	cos(27) = 0.891007
sin(28) = 0.469472	cos(28) = 0.882948
fx sin(29) = 0.484810	cos(29) = 0.874620

```
Command Window
sin(66) = 0.913545    cos(66) = 0.406737
sin(67) = 0.920505    cos(67) = 0.390731
sin(68) = 0.927184    cos(68) = 0.374607
sin(69) = 0.933580    cos(69) = 0.358368
sin(70) = 0.939693    cos(70) = 0.342020
sin(71) = 0.945519    cos(71) = 0.325568
sin(72) = 0.951057    cos(72) = 0.309017
sin(73) = 0.956305    cos(73) = 0.292372
sin(74) = 0.961262    cos(74) = 0.275637
sin(75) = 0.965926    cos(75) = 0.258819
sin(76) = 0.970296    cos(76) = 0.241922
sin(77) = 0.974370    cos(77) = 0.224951
sin(78) = 0.978148    cos(78) = 0.207912
sin(79) = 0.981627    cos(79) = 0.190809
sin(80) = 0.984808    cos(80) = 0.173648
sin(81) = 0.987688    cos(81) = 0.156434
sin(82) = 0.990268    cos(82) = 0.139173
sin(83) = 0.992546    cos(83) = 0.121869
sin(84) = 0.994522    cos(84) = 0.104528
sin(85) = 0.996195    cos(85) = 0.087156
sin(86) = 0.997564    cos(86) = 0.069756
sin(87) = 0.998630    cos(87) = 0.052336
sin(88) = 0.999391    cos(88) = 0.034899
sin(89) = 0.999848    cos(89) = 0.017452
sin(90) = 1.000000    cos(90) = 0.000000
fx >>
```

**Q-4.** Write a program to read a set of integers from an input data file, and locate the largest and smallest values within the data file. Print out the largest and smallest values, together with the lines on which they were found. Assume that you do not know the number of values in the file before the file is read.

**Code:**

```
clc;
clear all;
close all;

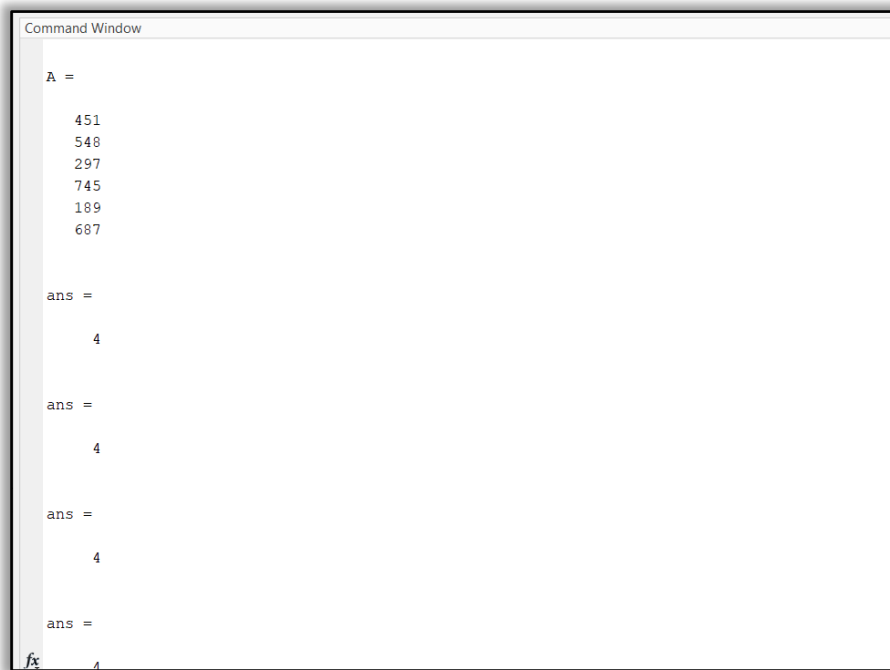
file=fopen('input.txt', 'wt');
```

✓ A=randi(1000, round(rand()\*10), 1)

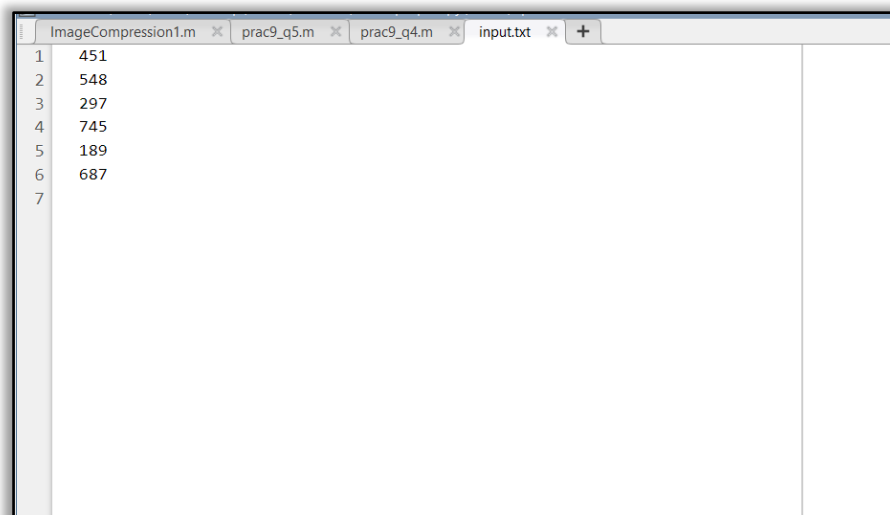
```
for ii=1:length(A)
    fprintf(file, '%d\n', A(ii))
end
```

```
fclose(file);  
  
fileID=fopen('input.txt', 'r');  
  
A=fscanf(fileID, '%d');  
  
fclose(fileID);
```

### Output:



A screenshot of the MATLAB Command Window showing the output of the program. The output consists of four lines: 'A =' followed by a column vector of six numbers (451, 548, 297, 745, 189, 687), 'ans =' followed by the number 4, 'ans =' followed by the number 4, and 'ans =' followed by the number 4. The Command Window title bar reads 'Command Window'.



A screenshot of the MATLAB Editor showing the 'input.txt' file. The file contains six lines of numbers: 451, 548, 297, 745, 189, and 687. The Editor title bar shows several open files: 'ImageCompression1.m', 'prac9\_q5.m', 'prac9\_q4.m', and 'input.txt'. The 'input.txt' file is currently selected and displayed in the main editor area.

**Q-5.** Angles are often measured in degrees ( $^{\circ}$ ), minutes ( $'$ ), and seconds ( $''$ ), with 360 degrees in a circle, 60 minutes in a degree, and 60 seconds in a minute. Write a program that reads angles in



radians from an input disk file and converts them into degrees, minutes, and seconds. Test your program by placing the following four angles expressed in radians into an input file and reading that file into the program: 0.0, 1.0, 3.141593, 6.0.

### Code:

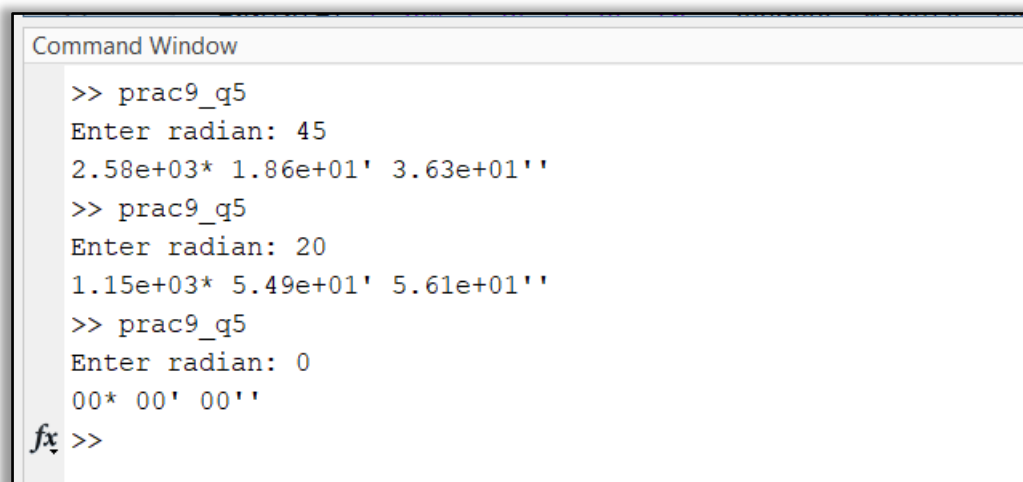
```
% clc;
clear all;
close all;

r=input("Enter radian: ");
degree=r*180/pi;

minute=rem(degree,1)*60;
second=rem(minute,1)*60;

fprintf("%.2d* %.2d' %.2d' '\n", degree, minute, second);
```

### Output:



```
Command Window

>> prac9_q5
Enter radian: 45
2.58e+03* 1.86e+01' 3.63e+01''
>> prac9_q5
Enter radian: 20
1.15e+03* 5.49e+01' 5.61e+01''
>> prac9_q5
Enter radian: 0
00* 00' 00''
fx >>
```

### Conclusion:

✓ From this experiment we learnt a lot about how to handle files and write and read input and outputs from files. We got to know the machinery that is libraries needed to work with files.