Lesson : 08 Use of `input`, `fprintf` and `sscanf` functions

Try the following exercises for understanding the use of `fprintf` function and these exercises need not be printed for record purpose

The **`fprintf`** Function

It is used for writing the formatted data to a file or on screen similar to **`fprintf`** in `C` language. However, we shall discuss here how **`fprintf`** can produce formatted output on display screen. Try the following in MATLAB command prompt.

ILLUSTRA TION

`‘%d’` prints a decimal integer in its position. Let us take an array `x` defined below

```
>>x=[12 4 32 15 20 7 13];
>> fprintf(‘%d’,x) % with no spaces between elements 124321520713
>> fprintf(‘%2d’,x)% with 12 4321520 713 2 spaces between
elements of the array x
>> fprintf(‘%3d’,x)% with 3 spaces between elements 12 4 32 15 20
7 13
```

`‘%o’`Converts a decimal array into octal array before printing on the screen

```
>> fprintf(‘%o’,x)% with no spaces between each element
144401724715
>> fprintf(‘%2o’,x)% with 2 spaces between each element 14
4401724 715
>> fprintf(‘%3o’,x) 14 4 40 17 24 7 15
```

`‘%x’`Converts a decimal array into hex array before printing on the screen >>

```
fprintf(‘%3x’,x)
c 4 20 f 14 7 d
```

`‘%f’`Converts the integers into floating points and prints. By default it displays in six-digit precision. However, we can change the precision digit by specifying a number after decimal point.

```
>> fprintf(‘%f’,x) % default precision of six digits.
12.0000004.00000032.00000015.00000020.0000007.00000013.000000
>> fprintf(‘%4.2f’,x)% the precision is set to 2 decimal places
12.004.0032.0015.0020.007.0013.00
>> fprintf(‘%5.2f’,x) 12.00 4.0032.0015.0020.00 7.0013.00
>> fprintf(‘%6.2f’,x)
12.0 4.00 32.00 15.00 20.00 7.00 13.00
```

The precision is 2 and the width 6 is enough to give correct output. `‘%g’` gives the output in integer or floating point of scientific notation

```
>> fprintf(‘%10.2g’,x) % gives integer output 75 45 93 47 42 85
53

>> fprintf(‘%10.4g’,x) % gives floating point output 74.68 44.51
93.18 46.6 41.86 84.62 52.52
```

`‘%c’` can be used to print a character or a character array. Let us take the character array string,

```
>> str=’karambi’ % creating a string or character array.
>> fprintf(’%c’,str) % prints no spaces between characters.
Karambi
>> fprintf(’%2c’,str) % prints with 2 spaces for each characters.
Karambi
```

Exercise: 01

The use of **`fprintf`** for a simple interest program.

Create a script file to calculate a simple interest

```
% SimpleInt1.m
% the following three variables p, r and m must be defined
% in the command window before running this program.
% Principal, p
% Rate of interest per month, r
% Number of months, m
I = (p*r*m)/100;
Amt = p+I;
fprintf(‘ Interest rate of %d at the rate of %d \n’,p,r);
fprintf(‘ per month after %d month is: %d \n’,m,I);
fprintf(‘ And the amount becomes from %d to %d’ p,Amt);
```

Save the program as `SimpleInt1.m` and let us specify the same values of p, r and m.

```
>>SimpleInt1 Interest of 2500 at the rate of 4 percent
Per month after 5 month is: 500.
And the amount becomes from 2500 to 3000
```

Exercise: 02

Use of **`input`** and **`fprintf`** in the same simple interest program.

Let us rewrite the same simple interest program discussed in Exercise: 01 to make it interactive using `input` function

```
% SimpleInt2.m
% Program to find simple interest
p = input (‘Enter the Principal: ‘);
r = input (‘Enter the rate per month: ‘);
m = input (‘Enter the number of months: ‘);
I = (p*r*m)/100;
Amt = p + I;
fprintf(‘ Interest rate of %d at the rate of %d \n’,p,r);
fprintf(‘ per month after %d month is: %d \n’,m,I);
fprintf(‘ And the amount becomes from %d to %d’ p,Amt);
```

Save this program as `SimpleInt2.m` and run the same in command window.

Exercise: 03

The use of **`sscanf`** function

```
%crmat.m
% To create a matrix of desired size from an array input.
% Size of the array should be entered as array of two elements.
x=input(‘Enter an array: ‘);
x1=floor(x);
difference=x-x1;
str=num2str(x);
sz=input(‘Size of the matrix: ‘);
if any(difference)
x=sscanf(str,’%f’, sz);
else x=sscanf(str,’%d’, sz);
end
```

Save the program as `crmat.m` and run the program

```
>>crmat Enter an array: [3 10 2 5 8 12 32 5 13 2 7 9 14 4 7]
Size of the matrix: [4 3]
3    8    13
10   12   2
2    32   7
5    5    9
```