

## Exercise 3: (Suggested time 3 hours)

1. Write a Python function to find the Max from a given list of numbers.
2. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument. (Hint: Recursive function)
3. Write a Python function that takes a list and returns a new list with unique elements of the first list.  
Sample List : [1,2,3,3,3,3,4,5]  
Unique List : [1, 2, 3, 4, 5]
4. Write a program to print the documentation content mentioned at the beginning of the function  
Eg:

```
def my_add(arg1, arg2):  
    "This is the document for my_add"  
    result = arg1 + arg2  
    print "arithmetic: add = {}".format(result)  
    return result
```

The program should print the line “This is the document for my\_add”

Hint: Use the function's internal variable `__doc__`.

Eg: `my_add.__doc__`

## Exercise 3: (Continued, Suggested time 3 hours)

Create two sample module named 'arithmetic', 'minmax' and 'sq\_sqrt'

Arithmetic module must contains your implementations for add, sub, mul and div

Minmax module must contain your implementations for min and max

sq\_sqrt module must contain your implementations for square and sqrt functions

1. Try importing these modules into you code
2. Import either the entire module or only required functions

Create a sample package named 'my\_math' using these modules

1. Create the `__init__.py` file for the package
2. Use the package in your program

## Exercise 3: (Continued, Suggested time 3 hours)

The following functions are not robust, and does not handle any exceptions. Modify them to handle the required exceptions.

```
def example1():
    for i in range( 3 ):
        x = int( input( "enter a number: " ) )
        y = int( input( "enter another number: " ) )
        print( x, '/', y, '=', x/y )
```

Hint test conditions: What will happen if non numeric type is given as input or one of the input is 0

```
def printUpperFile( fileName ):
    file = open( fileName, "r" )
    for line in file:
        print( line.upper() )
    file.close()
```

Hint test conditions: What happens if the file is not present or if the file is present but the user does not have read permission. What happens if the file is empty.