Practice exercises: Week 9 (Exceptions)

1. Define a subprogram **sum2(a, b)** that returns the sum of a and b provided user inputs are digits only. Otherwise display those input errors by using Python's exception coding techniques. The sample run is here:

```
# main program of question(1)
print(sum2(x,y))
```

```
>>> %Run prac1.py
 x?3
 y?6
 9.0

>>> %Run prac1.py
 x?
 y?asdfadsf
 input error
 None
```

Answer:

```
def sum2(a,b):
    try:
        a=float(a)
        b=float(b)
        return(a+b)
    except Exception:
        print("input error")
        return
x=input("x?")
y=input("y?")

# main program
print(sum2(x,y))
```

2. Rewrite the above program with user defined message **"input error"** if the input is not a number/digit and prompts the user to enter input again. The error message must be displayed until user input is correct to print the sum of a and b, then the program ends. The sample run is here:

Answer:

```
def sum2(a,b):
    try:
        a=float(a)
        b=float(b)
        return(a+b)
```

```
    except Exception:
        print("input error")
        return

# main program
x=input("x?")
y=input("y?")
while(sum2(x,y)==None):
    x=input("x?")
    y=input("y?")
print(sum2(x,y))
```

3. Write a sub program **divide(x,y)** that performs the division operation where x and y must be float data type. if the user input for y is 0 then it must be caught by Python's Error class "**Division by Zero**". Similarly, if the user input is not converted by float for division, then display an error message "Can't ***convert the input value into float***". [Hint: try-except→ ValueError structure]. The sample run is here:

```
>>> %Run prac2.py
 Input divisor:6
 Input dividend:a
 Can't convert input value to float!
>>> %Run prac2.py
 Input divisor:a
 Can't convert input value to float!
>>> %Run prac2.py
 Input divisor:9
 Input dividend:0
 division by zero!
>>> %Run prac2.py
 Input divisor:9
 Input dividend:3
 result is 3.0
```

Answer:

```
def divide():
    try:
        x=float(input("Input divisor:"))
        y=float(input("Input dividend:"))
        result = x / y
    except ZeroDivisionError:
        print("division by zero!")
```

```
    except ValueError:
        print("Can't convert input value to float!")
    else:
        print("result is", result)

#***main program***
divide()
```

4. Define a subprogram **ReadDataFile(filename)**, where file name is an argument. It should display an error message "**Can't find file**" if the file does not exist. Else, print the details stored in that file. Define custom exception to perform this [Slides 6-9]. The sample run is here:

```
>>> %Run prac3.py

  Please input file name to open.LibRec
  Can't find file.

>>> %Run prac3.py

  Please input file name to open.LibRec.txt
  Vivian   210001   21/10/2002
  Nancy    210002   7/8/2003
  Wilson   210003   27/2/2003
  James    210004   8/3/2004
  Bruce    210005   28/10/2002
```

```
def ReadDataFile(filename):
    try:
        f1=open(filename)
    except Exception as e:
        print("Can't find file:"+filename)
    else:
        filedata=f1.readlines()
        for x in filedata:
            print(x.strip())

#main program
filename=input("Please input file name to open.")
ReadDataFile(filename)
```