

1) Write a Python program which accepts a list named :

```
randomList = ['a',0,2].
```

Use exception handling using try-catch which gives the output as:

Output

1) If the List element is a alphabet or string, the output will be
The entry is a
Oops! <class 'ValueError'> occurred.

Next entry.

2) If the List element is "0", the output will be
The entry is 0
Oops! <class 'ZeroDivisionError'> occurred.

Next entry.

3) If the List element is an integer except 0, then output will be
The entry is 2
The reciprocal of 2 is 0.5 // reciprocal of an integer

In [41]:

```
def list_chk(randomList):
    for i in randomList:
        try:
            if str(i).isalpha():
                raise ValueError("The entry is : {a} - Oops! <class 'ValueError'> occurred")
            elif str(i) == '0':
                raise ZeroDivisionError("The entry is : 0 - Oops! <class 'ZeroDivisionError'> occurred")
            else:
                print("The entry is :",i)
                print("Reciprocal of {a} is: {b}".format(a=i,b=1/i))
        except ValueError as ex:
            print(ex)
        except ZeroDivisionError as ex1:
            print(ex1)
```

In [42]:

```
list_chk([0,'abc',5])
```

The entry is : 0 - Oops! <class 'ZeroDivisionError'> occurred.
The entry is : abc - Oops! <class 'ValueError'> occurred.
The entry is : 5
Reciprocal of 5 is: 0.2

In [40]:

```
list_chk([1,2,5])
```

```
The entry is : 1
Reciprocal of 1 is: 1.0
The entry is : 2
Reciprocal of 2 is: 0.5
The entry is : 5
Reciprocal of 5 is: 0.2
```

2) Array out of Bound Exception

Write a Python program to give exception "Array Out of Bound" if the user wants to access the elements beyond the list size (use try and except)

In [21]:

```
def chk_array_index(list,index):
    try:
        if index> len(list):
            raise IndexError
        else:
            print("In the given list the value of index {a} is : {b}".format(a=index,b=list[index]))
    except IndexError as ex:
        print("oops - Array out of bound")
        print(ex)
```

In [22]:

```
chk_array_index([1,2,3],3)
```

```
oops - Array out of bound
list index out of range
```

In [23]:

```
chk_array_index([1,2,3],1)
```

```
In the given list the value of index 1 is : 2
```

In [24]:

```
chk_array_index([1,2,3],2)
```

```
In the given list the value of index 2 is : 3
```

3) Write a python module script that contains fib2() method to calculate the fibonacci series till 1000 and save it as fibo.py.

Note : The module created as fibo.py has to be placed in lib folder

In [25]:

```
import fibo
```

In [26]:

```
fibo.fib2(-1)
```

Please enter a number greater than zero.

In [27]:

```
fibo.fib2(0)
```

Please enter a number greater than zero.

In [28]:

```
fibo.fib2(1)
```

Fibonacci sequence upto 1 :1

In [29]:

```
fibo.fib2(2)
```

Displaying Fibonacci sequence upto 2 :
0 , 1 , 1 , 2

In [30]:

```
fibo.fib2(5)
```

Displaying Fibonacci sequence upto 5 :
0 , 1 , 1 , 2 , 3 , 5

In [31]:

```
fibo.fib2(50)
```

Displaying Fibonacci sequence upto 50 :
0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34

In [32]:

```
fibo.fib2(1000)
```

Displaying Fibonacci sequence upto 1000 :
0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 , 144 , 233 , 377 , 610 ,
987

4) Write a python module script that contains ispalindrome() method to calculate the input string as palindrome string or not and save it as palindrome.py

In [33]:

```
import palindrome as p
```

In [34]:

```
p.ispalindrome('aba')
```

Yes it is Palindrome !

In [35]:

```
p.ispalindrome('abaaba')
```

Yes it is Palindrome !

In [36]:

```
p.ispalindrome('GeeksforGeeks')
```

Nope - it is not a Palindrome !

In [37]:

```
p.ispalindrome('Sudipta')
```

Nope - it is not a Palindrome !