Exercise

1

Write a function `division()` that accepts two arguments `x` and `y` and return the value of `x/y`.

The function must be able to catch an exception such as `ZeroDivisionError`, `ValueError`, or any unknown error you might come across when you are doing the division operation.

In [2]:

```
def division(x, y):
    pass
```

In [6]:

```
# This is some test cases. Run this cell to check!

for x in range(-2, 3):
    for y in range(-2, 3):
        z = division(x, y)
        print(f'{x}/{y} = {z}')
print('Test done.')
```

```
-2/-2 = None
-2/-1 = None
-2/0 = None
-2/1 = None
-2/2 = None
-1/-2 = None
-1/-1 = None
-1/0 = None
-1/1 = None
-1/2 = None
0/-2 = None
0/-1 = None
0/0 = None
0/1 = None
0/2 = None
1/-2 = None
1/-1 = None
1/0 = None
1/1 = None
1/2 = None
2/-2 = None
2/-1 = None
2/0 = None
2/1 = None
2/2 = None
Test done.
```

In [5]:

```python
# This is another test cases. Run this cell to check!

for x in [11, 22.5, -44, 'Hello', -11.25]:
    for y in range(-5, 5):
        z = division(x, y)
        print(f'{x}/{y} = {z}')
print('Test done.')
```

```
11/-5 = None
11/-4 = None
11/-3 = None
11/-2 = None
11/-1 = None
11/0 = None
11/1 = None
11/2 = None
11/3 = None
11/4 = None
22.5/-5 = None
22.5/-4 = None
22.5/-3 = None
22.5/-2 = None
22.5/-1 = None
22.5/0 = None
22.5/1 = None
22.5/2 = None
22.5/3 = None
22.5/4 = None
-44/-5 = None
-44/-4 = None
-44/-3 = None
-44/-2 = None
-44/-1 = None
-44/0 = None
-44/1 = None
-44/2 = None
-44/3 = None
-44/4 = None
Hello/-5 = None
Hello/-4 = None
Hello/-3 = None
Hello/-2 = None
Hello/-1 = None
Hello/0 = None
Hello/1 = None
Hello/2 = None
Hello/3 = None
Hello/4 = None
-11.25/-5 = None
-11.25/-4 = None
-11.25/-3 = None
-11.25/-2 = None
-11.25/-1 = None
-11.25/0 = None
-11.25/1 = None
-11.25/2 = None
-11.25/3 = None
-11.25/4 = None
Test done.
```

2

Write the guessing number game! Here is the rule:

- The player have to guess the correct number which is randomly chosen by the computer.
- At the beginning, computer choose an integer number randomly between intergers `a` and `b`. This is the number that the player has to guess in this game.
- The computer lets the player guesses until getting the correct number.

In this game, you have to handle some stupid inputs like guessing too low (less than `a`) or too high (greater than `b`) numbers with raising an error then let the player plays continue to play game

In [ ]:

```
import random

min_number = 0
max_number = 100

the_correct_number = random.randint(min_number, max_number)

game_run = True
# print(the_correct_number) # uncomment this for cheating the game

print('Welcome to the gussing game')
while game_run:
    x = int(input('Enter your guessing: '))
    if x == the_correct_number:
        print('Congratulations! Your guessing is correct')
        game_run = False
    else:
        print(f'Try again, it is not {x}')

print('Thanks for playing')
```

3

Write a program that prompts the user for a number, and then calculates the square of that number. If the user enters a value that is not a number (such as a letter or symbol), the program should display an error message and prompt the user to enter a valid number again.

In [ ]:

```
# Fix this script

num = float(input("Enter a number: "))
result = num ** 2
print(f'The square of {num} is {result}')
```

4

Create a program that reads a file and prints the contents of the file to the screen. If the file does not exist or if the file cannot be opened for reading, the program should display an appropriate error message and exit.

In [ ]:

```python
# Fix this script

with open("example.txt", "r") as file:
    for line in file:
        print(line)
```