Lab: Exception Handling

1. Explain each exception type given below and write examples of each:

a. NameError

b. ValueError

c. AttributionError

d. IndexError

e. KeyError

f. SyntaxError

g. IOError

h. ImportError

i. ZeroDivisionError

2. Add a try/except clause to run the code without errors. If key error then you should add a key with value 0 i.e. If a blog post didn't get any likes, a 'Likes' key should be added to that dictionary with a value of 0.

```
blog_posts = [{'Photos': 3, 'Likes': 21, 'Comments': 2}, {'Likes': 13, 'Comments':
2, 'Shares': 1}, {'Photos': 5, 'Likes': 33, 'Comments': 8, 'Shares': 3}, {'Comments':
4, 'Shares': 2}, {'Photos': 8, 'Comments': 1, 'Shares': 1}, {'Photos': 3, 'Likes': 19,
'Comments': 3}]

total_likes = 0

for post in blog_posts:

   total_likes = total_likes + post['Likes']
```

3. The code below assigns the 5th letter of each word in food to the new list **fifth**. If the fifth letter is not available then make provision in try/except clause to add letter N to the list.

```
food = ["chocolate", "chicken", "corn", "sandwich", "soup", "potatoes", "beef", "lox", "lemonade"]

fifth = []

for x in food:

   fifth.append(x[4])
```

4. Write a python script to receive two lists at run time and store the sum of the corresponding elements of both the lists in a new list. Use try/except clause to address possible errors.

5. Write a python script to receive two numbers at run time and add these numbers and print remainder when this addition is divided with the difference of received numbers. Use try/except clause to address possible errors.