

Assignment-3

Task 1:

1.

Write a function to compute 5/0 and use try/except to catch the exceptions.

```
In [1]: #function for compute 5/0
def compute_5by0():
    return 5/0
# try block
try :
    compute_5by0()

#except block
except ZeroDivisionError as e:
    print("5 can't be devide by zero. Exception occured as ' {} ' ".format(e))

except:
    print("There is another exception error")
```

5 can't be devide by zero. Exception occured as ' division by zero ' .

2-

Implement a Python program to generate all sentences where subject is in ["Americans", "Indians"] and verb is in ["Play", "watch"] and the object is in ["Baseball", "cricket"].

Hint: Subject, Verb and Object should be declared in the program as shown below.

```
subjects=["Americans ", "Indians"]
verbs=["play", "watch"]
objects=["Baseball", "Cricket"]
```

Output should come as below:

```
Americans play Baseball.
Americans play Cricket.
Americans watch Baseball.
Americans watch Cricket.
Indians play Baseball.
Indians play Cricket.
Indians watch Baseball.
Indians watch Cricket.
```

```
In [3]: subjects=["Americans ", "Indians"]
verbs=["play", "watch"]
objects=["Baseball", "Cricket"]

# creation of all sentences by List comprehension.
all_sentence=[(s+" "+v+" "+o+".") for s in subjects for v in verbs for o in objects ]

#for Loop to print all created sentences one by one.
for sentence in all_sentence:
    print(sentence)
```

```
Americans play Baseball.
Americans play Cricket.
Americans watch Baseball.
Americans watch Cricket.
Indians play Baseball.
Indians play Cricket.
Indians watch Baseball.
Indians watch Cricket.
```

Task-2

1. Write a function so that the columns of the output matrix are powers of the input vector. The order of the powers is determined by the increasing boolean argument. Specifically, when increasing is False, the i-th output column is the input vector raised element-wise to the power of $N - i - 1$.
HINT: Such a matrix with a geometric progression in each row is named for Alexandre-Theophile Vandermonde.

```
In [7]: import numpy as np
def Vandermonde_func(input_vector, N, increasing=False):
    if increasing == False:
        output = np.array([x**(N-1-i) for x in input_vector for i in range(N)]).reshape(len(input_vector),N)
    elif increasing == True:
        output = np.array([x**i for x in input_vector for i in range(N)]).reshape(len(input_vector),N)
    return output

input_vector = np.array([1,2,3,4,5])
N=5

print("When increasing is True\n", Vandermonde_func(input_vector, N, increasing=True))
print("When increasing is False\n",Vandermonde_func(input_vector, N, increasing=False))

When increasing is True
[[ 1  1  1  1  1]
 [ 1  2  4  8 16]
 [ 1  3  9 27 81]
 [ 1  4 16 64 256]
 [ 1  5 25 125 625]]
When increasing is False
[[ 1  1  1  1  1]
 [16  8  4  2  1]
 [81 27  9  3  1]
 [256 64 16  4  1]
 [625 125 25  5  1]]
```