Task 1:

1.

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

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.

1

In [1]:

```python
def divbyzero(x):
    try:
        print (x/0)
    except Exception as e:
        print (e)
```

In [2]:

```python
divbyzero(5)
```

```
division by zero
```

2

In [3]:

```python
def sentence(s, v, o):
    for i in s:
        for j in v:
            for k in o:
                print(i,j,k)
```

In [4]:

```python
subjects=["Americans ","Indians"]

verbs=["play","watch"]

objects=["Baseball","Cricket"]

sentence(subjects, verbs, objects)
```

```
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:

In [ ]:

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.

In [18]:

```
import numpy as np

def Alex_Theo_Vand(inputV, N, increasing):
    inputV=np.array(inputV)
    if not increasing:
        return np.column_stack([inputV**(N-i-1) for i in range(N)])
    else:
        return np.column_stack([inputV**i for i in range(N)])
```

In [20]:

```
inputV = [4, 5, 6, 7, 8]
N = 5
increasing = True


Alex_Theo_Vand(inputV, N, increasing)
```

Out[20]:

```
array([[   1,    4,   16,   64,  256],
       [   1,    5,   25,  125,  625],
       [   1,    6,   36,  216, 1296],
       [   1,    7,   49,  343, 2401],
       [   1,    8,   64,  512, 4096]], dtype=int32)
```

In [ ]:

In [ ]:

In [ ]: