# Task 1

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

In [5]:

```
def divide(numerator , denominator):
    try:
        return numerator/denominator
    except:
        print('error while dividing the numerator by denominator')
```

```
error while dividing the numerator by denominator
```

In [6]:

```
divide(5,0)
```

```
error while dividing the numerator by denominator
```

### 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"]

In [10]:

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

for sentence in ['{0} {1} {2}'.format(subject, verb, obj) for subject in subjects
    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-thoutput column is the input vector raised element-wise to the power of N - i - 1.

HINT:Such amatrix with ageometric progression in each row is named for Alexandre-Theophile Vandermonde.

In [19]:

```
def vandermonde_matrix(x, N=None, increasing = False):

    if N == None : N = len(x)
    if increasing:
        return [ [element**p for p in range(0, N)] for element in x]
    else:
        return [ [element**p for p in range(N-1, -1, -1)] for element in x]
```

In [20]:

```
print ( vandermonde_matrix([1, 2, 3, 5]))
```

```
[[1, 1, 1, 1], [8, 4, 2, 1], [27, 9, 3, 1], [125, 25, 5, 1]]
```

In [21]:

```
print ( vandermonde_matrix([1, 2, 3, 5], N=4 ) )
```

```
[[1, 1, 1, 1], [8, 4, 2, 1], [27, 9, 3, 1], [125, 25, 5, 1]]
```

In [22]:

```
print ( vandermonde_matrix([1, 2, 3, 5], N=4, increasing=True ) )
```

```
[[1, 1, 1, 1], [1, 2, 4, 8], [1, 3, 9, 27], [1, 5, 25, 125]]
```