assignment3

May 13, 2020

Task 1:

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

```python
[5]: def div_by_zero(a, b):
         try:
             return a / b
         except ZeroDivisionError as e:
             print(e)
```

```python
[6]: div_by_zero(5, 0)
```

```
division by zero
```

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"]

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

      import itertools

      for s in subjects:
          for v in verbs:
              for o in objects:
                  sentence = " ".join([s, v, o,]) + '.'
                  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 pow
of N - i - 1.
HINT: Such a matrix with a geometric progression in each row is named for Alexandre-
Theophile Vandermonde.

```python
[30]: def vandermonde(vector, increase=True):
          m = []
          for v in vector:
              row = [v ** i for i in range(len(vector))]
              if not increase:
                  row.reverse()
              m.append(row)
          return(m)
```

```python
[31]: v = [1, 2, 3, 4]
      print(*vandermonde(v), sep='\n')
      print(*vandermonde(v, increase=False), sep='\n')
```

```
[1, 1, 1, 1]
[1, 2, 4, 8]
[1, 3, 9, 27]
[1, 4, 16, 64]
[1, 1, 1, 1]
[8, 4, 2, 1]
[27, 9, 3, 1]
[64, 16, 4, 1]
```