

Session 3

Assignment 1

2. Problem Statement

Task 1:

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

```
a=5
try:
    a=a/0
    print("Done")
except Exception as e:
    print(e," this is the run time error.")
```

Output:

```
division by zero this is the run time error.
```

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.

```

main=[]
subject = ["Americans", "Indians"]
verb = ["Play", "watch"]
objects = ["Baseball", "cricket"]

for i in subject:
    for j in verb:
        for k in objects:
            main.append(i+" "+j+" "+k)

for i in main:
    print(i)

```

Output:

```

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.

```

import numpy as np

def myMatrixFunction(pVector, n, booleanValueIn):

    if not booleanValueIn:
        temp = []
        for x in pVector:
            for i in range(n):
                temp.append(x**(n-1-i))
            giveBack = np.array(temp).reshape(pVector.size,n)

    elif booleanValueIn:
        temp = []
        for x in pVector:
            for i in range(n):
                temp.append(x**i)
            giveBack = np.array(temp).reshape(pVector.size,n)

    return giveBack

temp = list(input("Enter the Interger List ").split())
temp = list(map(int, temp))
userVector = np.array(temp)
noOfColumn = int(input("Enter the Numbers of Columns "))
while True:
    booleanValue = int(input("Enter 0 for false boolean value \n Enter 1 for ture
boolean value.\n Please Enter your choice "))
    print("The input array is:",userVector,"\n")
    if booleanValue==0:
        falseVector = myMatrixFunction(userVector,noOfColumn,False)
        print("The Boolean value is FALSE Hence Matrix is \n\n",falseVector,"\n"
)
        break
    elif booleanValue==1:
        tureVector = myMatrixFunction(userVector,noOfColumn,True)
        print("The Boolean value is TRUE Hence Matrix is \n\n",tureVector,"\n")
        break

```

Output:

Enter the Interger List 1 2 5 4 3 6

Enter the Numbers of Columns 5

Enter 0 for false boolean value

Enter 1 for ture boolean value.

Please Enter your choice 1

The input array is: [1 2 5 4 3 6]

The Boolean value is TRUE Hence Matrix is

```
[[ 1  1  1  1  1]
 [ 1  2  4  8 16]
 [ 1  5 25 125 625]
 [ 1  4 16  64 256]
 [ 1  3  9  27  81]
 [ 1  6 36 216 1296]]
```