**Assignment 1**

**MET CS 521**

**Information structures with python**

**Purpose**

To construct a program to display the correct President Lincoln’s Gettysburg Address.

**Input :**

I used the \_init\_.py file as input ,

First load the file into the program and then match.

**Code/Comment:**

I have attached the give\_gettysburg\_test.py. which contains and code and comment .

**-----------------------------------Code---------------------**

from Mishra\_neha\_assignment\_1 import gettysburg\_

# === (Introduced): Single lines are on the console introducing

# the application, the fragment count, and instructions respectively

print("\nWelcome to LEARNING THE GETTYSBURG ADDRESS")

print("The Address has been broken into {0} fragments".format(str(len(gettysburg\_))))

print("\nReconstruct beginning of Gettysburg Address by accepting or rejecting:")

# === (Choices Appeared): After each input, all user's prior choices are on console

# === (Selection Recorded):

# User has selected gettysburg\_[i] for all i in selection\_

# AND selection\_ contains no repeats

# --IS FULFILLED TOGETHER WITH--

# === (Next to Display):

# gettysburg\_[next\_] is to be displayed after the current one

# XOR len(selection\_) = len(gettysburg\_)

selected\_fragments = []

display\_order = [3, 4, 1, 5, 2, 0]

display\_order\_index = 0

while len(selected\_fragments) < len(gettysburg\_):

print("\nSELECT THE FOLLOWING NEXT? ('y' otherwise any other character): \n" + gettysburg\_[

display\_order[display\_order\_index]])

response\_ = raw\_input() #because I cannot take input directly

# Restore "Selection Recorded"

if response\_ == 'y':

selected\_fragments.append(display\_order[display\_order\_index])

del display\_order[display\_order\_index] #

# Restore "Next to Display"

display\_order\_index = 0

print("\n----------------------YOUR CHOICES SO FAR, IN ORDER----------------------")

elif response\_ != 'y':

print("\n----------------------YOUR CHOICES SO FAR, IN ORDER----------------------")

# Restore "Next to Display"

display\_order\_index += 1

# ===(Selection Displayed):User has selected gettysburg\_[i] for all i in selected\_fragments AND selected\_fragments contains no repeats.

for i in selected\_fragments:

print(gettysburg\_[i])

the code is attached also

**Output :**

When we run the code is arranged in the order [3, 4, 1, 5, 2, 0], so the output is shown as below:









