Experiment 11

Experiment: Document Summarization Bot.

Aim : Write a Program for Document Summarization Bot.

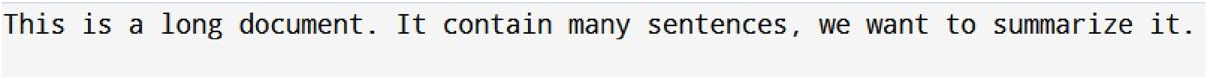
Program :

```
def summarize_text(text, n=2):
    sentences = text.split(".")
    summary = ".".join(sentences[:n]) + "."
    return summary


text = "This is a long document. It contain many sentences, we want to summarize it"
summary = summarize_text(text)
print(summary)
```

Input :

"This is a long document. It contain many sentences, we want to summarize it"

Output :



```
This is a long document. It contain many sentences, we want to summarize it.
```

Result : Thus, the program has been successfully summarize documents and verified.

Experiment 12

Experiment: Voice Command Automation (speech-to-text).

Aim : Write a Program for Voice Command Automation.

Program :

```
import speech_recognition as sr

def speech_to_text(audio_file):

    recognizer = sr.Recognizer()

    with sr.AudioFile(audio_file) as source:

        audio = recognizer.record(source)  # Properly indented

    text = recognizer.recognize_google(audio)

    return text

audio_file = "Audio.wav"

text = speech_to_text(audio_file)

print(text)
```

Input :

Audio.wav (audio file)

Output :

```
This file contains a sequence of pure tones at various frequency designed for audio system

testing provided by Sample files.
```

Result : Thus, the program has been successfully implemented voice command automation and verified.

Experiment 13

Experiment: Simple Fraud Detection in Transaction.

Aim : Write a Program for Simple Fraud Detection in Transaction.

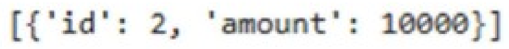
Program :

```
transactions = [

{"id" : 1, "amount" : 500},{"id" : 2, "amount" : 10000},{"id" : 3, "amount" : 200}

]

def detect_fraud(transactions, threshold=1000):

suspicious = [t for t in transactions if t["amount"] > threshold]

return suspicious

transaction_list = transactions

frauds = detect_fraud(transaction_list)

print(frauds)
```

Input :

```
[{"id" : 1, "amount" : 500},{"id" : 2, "amount" : 10000},{"id" : 3, "amount" : 200}]
```

Output :



```
[{'id': 2, 'amount': 10000}]
```

Result : Thus, the program has been successfully implemented Fraud Detection in Transaction and verified.

Experiment 14

Experiment: Inventory Alert System.

Aim : Write a Program for Inventory Alert System.

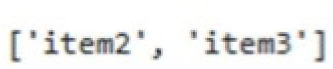
Program :

```
inventory = {

"item1" : 20,"item2" : 5,"item3" : 0

}

def check_inventory(inventory, min_stock=10):

alerts = [item for item, qty in inventory.items() if qty < min_stock]

return alerts

current_inventory = inventory

low_stock_items = check_inventory(current_inventory)

print(low_stock_items)
```

Input :

```
{"item1" : 20,"item2" : 5,"item3" : 0 }
```

Output :



```
['item2', 'item3']
```

Result : Thus, the program has been successfully implemented inventory alert system and verified.

Experiment 15

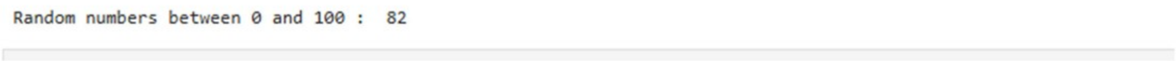
Experiment: Generate Random Number Between 0 to 100.

Aim : Write a Program to Generate random number between 0 to 100.

Program :

```
from import random

random_num = random.randint(0, 100)

print("Random number between 0 and 100 : ", random_num)
```

Input / Output :



```
Random numbers between 0 and 100 :  82
```

Result : Thus, the program has been successfully generated number and verified.

Experiment 16

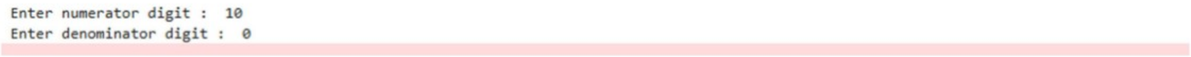
Experiment: Demonstrate an Aim to implement Abnormal Contribution

Aim : Write a Program to demonstrate an aim to implement Abnormal Contribution.

Program :

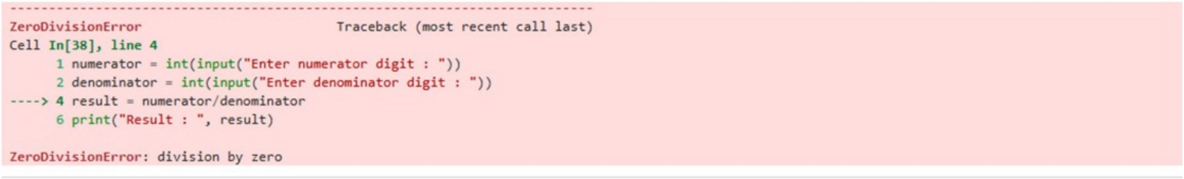
```
numerator = int(input("Enter numerator digit : "))

denominator = int(input("Enter denominator digit : "))

result = numerator/denominator

print("Result : ", result)
```

Input :



```
Enter numerator digit :  10
Enter denominator digit :  0
```

Output :



```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[38], line 4
      1 numerator = int(input("Enter numerator digit : "))
      2 denominator = int(input("Enter denominator digit : "))
----> 4 result = numerator/denominator
      6 print("Result : ", result)

ZeroDivisionError: division by zero
```

Result : Thus, the program has been successfully implemented Abnormal Contribution and verified.