```
                                                                         #
1.Mad Libs

print("GOUTHAM KB  \n1AY24AI035 \nAIML 'M' ")
import re
def mad_libs(python):
    with open(python, 'r') as file:
        text = file.read()
    placeholders = re.findall(r'\b(ADJECTIVE|NOUN|VERB|ADVERB)\b',
text)
    user_inputs = []

    for word in placeholders:
        article = 'an' if word[0] in 'AEIOU' else 'a'
        user_input = input(f"Enter {article.lower()} {word.lower()}:
")
        user_inputs.append(user_input)

    def replace_match(match):
        return user_inputs.pop(0)
    result_text = re.sub(r'\b(ADJECTIVE|NOUN|VERB|ADVERB)\b',
replace_match, text)
    print("\n--- Final Mad Libs Story ---")
    print(result_text)
    output_filename = 'python.txt'
    with open(output_filename, 'w') as output_file:
        output_file.write(result_text)

    print(f"\nThe completed story has been saved to
'{output_filename}'.")
mad_libs('python.txt')

GOUTHAM K B
1AY24AI035
AIML 'M'

--- Final Mad Libs Story ---
The silly panda walked to the chandelier and then screamed. A nearby
pickup truck was unaffected by these events.


The completed story has been saved to 'python.txt'.




                                                              # 2.Regex
Search

print("GOUTHAM KB  \n1AY24AI035 \nAIML 'M' ")
```

```
import os
import re

def regex_search_in_files(folder_path
                         ):
    pattern = input("Enter the regular expression to search for: ")
    try:
        regex = re.compile(pattern)
    except re.error:
        print("Invalid regular expression.")
        return
    for filename in os.listdir(folder_path):
        if filename.endswith(".txt"):
            filepath = os.path.join(folder_path, filename)
            with open(filepath, 'r', encoding='utf-8') as file:
                lines = file.readlines()
                for line_num, line in enumerate(lines, start=1):
                    if regex.search(line):
                        print(f"{filename} [Line {line_num}]:
{line.strip()}")
folder = input("Enter folder path: ").strip()
if os.path.isdir(folder):
    regex_search_in_files(folder)
else:
    print("Invalid folder path.")
```

```
GOUTHAM KB
1AY24AI035
AIML 'M'

Enter folder path:  C:\Users\ANISH B L\OneDrive\Desktop\Python
Enter the regular expression to search for:  Virat

cricket.txt [Line 1]: Virat Kohli is an Indian international cricketer
who plays ODI cricket for the India national team and is a former
captain in all formats. He is a right-handed batsman and occasional
right-arm medium pace bowler.
```

```
                                                            # 3.
Selective Copy

print("GOUTHAM KB \n1AY24AI035 \nAIML 'M' ")
import os
import shutil

def selective_copy(src_folder, dest_folder, extension):
    extension = extension.lower().strip(".")
    if not os.path.exists(dest_folder):
        os.makedirs(dest_folder)
    for foldername, subfolders, filenames in os.walk(src_folder):
        for filename in filenames:
            if filename.lower().endswith("." + extension):
                src_path = os.path.join(foldername, filename)
                dest_path = os.path.join(dest_folder, filename)
                if not os.path.exists(dest_path):
                    shutil.copy(src_path, dest_path)
                    print(f"Copied: {src_path} → {dest_path}")
                else:
                    print(f"Skipped (already exists): {dest_path}")

src = input("Enter source folder path: ").strip()
dest = input("Enter destination folder path: ").strip()
ext = input("Enter file extension to search for (e.g., pdf, jpg): 
").strip()

if os.path.isdir(src):
    selective_copy(src, dest, ext)
else:
    print("Invalid source folder.")
```

```
GOUTHAM KB
1AY24AI035
AIML 'M'

Enter source folder path:  C:\Users\ANISH B L\OneDrive\Pictures\
Pictures
Enter destination folder path:  C:\Users\ANISH B L\OneDrive\Documents
Enter file extension to search for (e.g., pdf, jpg):  jpg

Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\HanuMan-8.jpg →
C:\Users\ANISH B L\OneDrive\Documents\HanuMan-8.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG20231113193630.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG20231113193630.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG20231113193659.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG20231113193659.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG20240501191523.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG20240501191523.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20221106_182626_050.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20221106_182626_050.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20230802_182211_267.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20230802_182211_267.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20230802_182213_740.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20230802_182213_740.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20231023_080217_727.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20231023_080217_727.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20231115_180319_193.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20231115_180319_193.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240225_100820_271.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240225_100820_271.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240225_100825_722.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240225_100825_722.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240313_141409_310.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240313_141409_310.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240319_195216_535.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240319_195216_535.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240511_150050_314.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240511_150050_314.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240603_222654_027.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240603_222654_027.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240630_081154_562.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240630_081154_562.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240630_110008_628.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240630_110008_628.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240704_145719_260.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240704_145719_260.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240712_222653_829.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240712_222653_829.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\
IMG_20240724_194109_460.jpg → C:\Users\ANISH B L\OneDrive\Documents\
IMG_20240724_194109_460.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\King Trophy.jpg
→ C:\Users\ANISH B L\OneDrive\Documents\King Trophy.jpg
Copied: C:\Users\ANISH B L\OneDrive\Pictures\Pictures\Shiva.jpg → C:\
Users\ANISH B L\OneDrive\Documents\Shiva.jpg


                                                             # 4. Deleting
Unneeded Files
```

```
print("GOUTHAM KB  \n1AY24AI035 \nAIML 'M' ")
import os

def find_large_files(folder_path, size_limit_mb=100):
    size_limit_bytes = size_limit_mb * 1024 * 1024
    print(f"\nFiles larger than {size_limit_mb} MB in
'{folder_path}':\n")

    for foldername, subfolders, filenames in os.walk(folder_path):
        for filename in filenames:
            try:
                filepath = os.path.join(foldername, filename)
                filesize = os.path.getsize(filepath)
                if filesize > size_limit_bytes:
                    size_in_mb = round(filesize / (1024 * 1024), 2)
                    print(f"{os.path.abspath(filepath)} - {size_in_mb}
MB")
            except (FileNotFoundError, PermissionError):
                continue
folder = input("
jnEnter folder path to scan: ").strip()
size_limit = input("Enter size limit in MB (default is 100):
").strip()

try:
    size_limit = int(size_limit) if size_limit else 100
    if os.path.isdir(folder):
        find_large_files(folder, size_limit)
    else:
        print("Invalid folder path.")
except ValueError:
    print("Size limit must be a number.")
```

```
GOUTHAM KB
1AY24AI035
AIML 'M'

Enter folder path to scan:  Downloads
Enter size limit in MB (default is 100):  1000


Files larger than 1000 MB in 'Downloads':

C:\Users\ANISH B L\Downloads\Arjuna Phalguna (2022)
1080p.UnCut.HDRip.Dual.Vegamovies.NL.mkv - 2333.25 MB
```

```
                                                     # 5. Filling
in the Gaps
```

```
print("GOUTHAM KB \n1AY24AI035 \nAIML 'M' "
)import os
import re

def fill_gaps(folder, prefix, extension):
    files = os.listdir(folder)
    pattern = re.compile(rf"^{re.escape(prefix)}(\d+)\.
{re.escape(extension)}$")

    matched_files = []
    for file in files:
        match = pattern.match(file)
        if match:
            number = int(match.group(1))
            matched_files.append((number, file))

    matched_files.sort()

    expected_number = 1
    for actual_number, filename in matched_files:
        if actual_number != expected_number:
            new_name = f"{prefix}{str(expected_number).zfill(3)}.
{extension}"
            print(f"Renaming {filename} → {new_name}")
            os.rename(
                os.path.join(folder, filename),
                os.path.join(folder, new_name)
            )
        expected_number += 1

# Example usage
folder = input("Enter folder path: ").strip()
prefix = input("Enter file prefix (e.g., spam): ").strip()
extension = input("Enter file extension (without dot, e.g., txt):
").strip()

if os.path.isdir(folder):
    fill_gaps(folder, prefix, extension)
else:
    print("Invalid folder path.")


                                                        # Inserting
gaps

import os
import re

def insert_gap(folder, prefix, extension, position):
    files = os.listdir(folder)
    pattern = re.compile(rf"^{re.escape(prefix)}(\d+)\.
{re.escape(extension)}$")

    matched_files = []
    for file in files:
        match = pattern.match(file)
        if match:
            number = int(match.group(1))
            matched_files.append((number, file))

    matched_files.sort(reverse=True)  # Reverse to rename from end to
avoid overwriting

    for number, filename in matched_files:
        if number >= position:
            new_number = number + 1
            new_name = f"{prefix}{str(new_number).zfill(3)}.
{extension}"
            print(f"Renaming {filename} → {new_name}")
            os.rename(
                os.path.join(folder, filename),
                os.path.join(folder, new_name)
            )

folder = input("\nEnter folder path: ").strip()
prefix = input("Enter file prefix (e.g., spam): ").strip()
extension = input("Enter file extension (without dot, e.g., txt):
").strip()
position = int(input("Enter position number to insert gap (e.g., 2):
").strip())

if os.path.isdir(folder):
    insert_gap(folder, prefix, extension, position)
else:
    print("Invalid folder path.")
```

GOUTHAM KB
1AY24AI035
AIML 'M'

```
Enter folder path:  C:\Users\ANISH B L\OneDrive\Desktop\Python
Enter file prefix (e.g., spam):  cricket
Enter file extension (without dot, e.g., txt):  txt

Renaming cricket003.txt → cricket002.txt


Enter folder path:  C:\Users\ANISH B L\OneDrive\Desktop\Python
Enter file prefix (e.g., spam):  cricket
Enter file extension (without dot, e.g., txt):  txt
Enter position number to insert gap (e.g., 2):  1

Renaming cricket002.txt → cricket003.txt
Renaming cricket001.txt → cricket002.txt
```

*# 6. Debugging Coin Toss*

```python
print("GOUTHAM KB  \n1AY24AI035 \nAIML 'M' ")
import random

guess = ''
while guess not in ('heads', 'tails'):
    print('Guess the coin toss! Enter heads or tails:')
    guess = input().lower()

toss = random.randint(0, 1)
toss_result = 'heads' if toss == 1 else 'tails'

if guess == toss_result:
    print('You got it!')
else:
    print('Nope! Guess again!')
    guess = input().lower()
    if guess == toss_result:
        print('You got it!')
    else:
        print('Nope. You are really bad at this game.')
```

```
GOUTHAM KB  
1AY24AI036
AIML 'M'
Guess the coin toss! Enter heads or tails:

 heads

You got it!
```

*# 7. Project 7*

```python
print("GOUTHAM KB\n1AY24AI035 \nAIML 'M' ")
import math

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

class Rectangle:
    def __init__(self, corner, width, height):
        self.corner = corner
        self.width = width
        self.height = height

class Circle:
    def __init__(self, center, radius):
        self.center = center
        self.radius = radius

def point_in_circle(circle, point):
    dx = point.x - circle.center.x
    dy = point.y - circle.center.y
    distance = math.hypot(dx, dy)
    return distance <= circle.radius

def rect_in_circle(circle, rect):
    corners = [
        rect.corner,
        Point(rect.corner.x + rect.width, rect.corner.y),
        Point(rect.corner.x, rect.corner.y + rect.height),
        Point(rect.corner.x + rect.width, rect.corner.y + rect.height)
    ]
    return all(point_in_circle(circle, corner) for corner in corners)

def rect_circle_overlap(circle, rect):
    corners = [
        rect.corner,
        Point(rect.corner.x + rect.width, rect.corner.y),
        Point(rect.corner.x, rect.corner.y + rect.height),
        Point(rect.corner.x + rect.width, rect.corner.y + rect.height)
    ]
    if any(point_in_circle(circle, corner) for corner in corners):
        return True
    if (rect.corner.x <= circle.center.x <= rect.corner.x + rect.width
and
        rect.corner.y <= circle.center.y <= rect.corner.y +
rect.height):
        return True
    return False

circle = Circle(Point(150, 100), 75)
rectangle = Rectangle(Point(120, 80), 40, 30)
point = Point(160, 110)
print("Point in Circle:", point_in_circle(circle, point))
print("Rectangle fully in Circle:", rect_in_circle(circle, rectangle))
print("Rectangle overlaps Circle:", rect_circle_overlap(circle,
rectangle))
```

```
Anish B L
1AY24AI010
AIML 'M'
Point in Circle: True
Rectangle fully in Circle: True
Rectangle overlaps Circle: True
```

```
                                                               # 8.
Project 8

print("GOUTHAM KB \n1AY24AI035 \nAIML 'M' ")
class Time:
    def __init__(self, hours=0, minutes=0, seconds=0):
        self.hours = hours
        self.minutes = minutes
        self.seconds = seconds
    def __str__(self):
        return f"{self.hours:02}:{self.minutes:02}:{self.seconds:02}"
    def to_seconds(self):
        return self.hours * 3600 + self.minutes * 60 + self.seconds
    def from_seconds(self, total_seconds):
        self.hours = total_seconds // 3600
        total_seconds %= 3600
        self.minutes = total_seconds // 60
        self.seconds = total_seconds % 60
        return self
def mul_time(time, number):
    total_seconds = time.to_seconds() * number
    result = Time().from_seconds(int(total_seconds))
    return result
def average_pace(finishing_time, distance):
    pace_seconds = finishing_time.to_seconds() / distance
    return Time().from_seconds(int(pace_seconds))

finish = Time(1, 30, 0)
distance = 10
print("Time multiplied by 2:", mul_time(finish, 2))
print("Average pace per mile:", average_pace(finish, distance))
```

```
GOUTHAM KB
1AY24AI035
AIML 'M'
Time multiplied by 2: 03:00:00
Average pace per mile: 00:09:00
```

```
                                                            # 9.
Project 9

print("GOUTHAM KB 1AY24AI0035 \nAIML 'M' ")

from datetime import datetime, timedelta
def show_today():
    today = datetime.today()
    print("Today's date:", today.date())
    print("Day of the week:", today.strftime("%A"))  # e.g., Monday
def birthday_info():
    bday_str = input("Enter your birthday (YYYY-MM-DD): ")
    birthday = datetime.strptime(bday_str, "%Y-%m-%d")
    today = datetime.today()
    age = today.year - birthday.year
    if (today.month, today.day) < (birthday.month, birthday.day):
        age -= 1
    print(f"You are {age} years old.")
    next_birthday = birthday.replace(year=today.year)
    if next_birthday < today:
        next_birthday = next_birthday.replace(year=today.year + 1)

    time_until = next_birthday - today
    total_seconds = int(time_until.total_seconds())
    days = total_seconds // 86400
    hours = (total_seconds % 86400) // 3600
    minutes = (total_seconds % 3600) // 60
    seconds = total_seconds % 60

    print(f"Time until your next birthday: {days} days, {hours} hours,
{minutes} minutes, {seconds} seconds")

def double_day(birth1_str, birth2_str):
    b1 = datetime.strptime(birth1_str, "%Y-%m-%d")
    b2 = datetime.strptime(birth2_str, "%Y-%m-%d")
    if b1 > b2:
        b1, b2 = b2, b1

    delta = b2 - b1
    double_day = b2 + delta
    print("Double Day is:", double_day.date())

def n_times_day(birth1_str, birth2_str, n):
    b1 = datetime.strptime(birth1_str, "%Y-%m-%d")
    b2 = datetime.strptime(birth2_str, "%Y-%m-%d")

    if b1 > b2:
        b1, b2 = b2, b1
    delta = b2 - b1
    n_day = b2 + delta / (n - 1)
    print(f"The day one person is {n} times older is:", n_day.date())
if __name__ == "__main__":
    print("\n1. Show Today's Date and Day of Week")
    show_today()

    print("\n2. Birthday Info")
    birthday_info()

    print("\n3. Double Day")
    double_day("2000-01-01", "2005-01-01")

    print("\n4. N-times Older Day")
    n_times_day("2000-01-01", "2005-01-01", 3)
```

```
GOUTHAM KB
1AY24AI035
AIML 'M'

1. Show Today's Date and Day of Week
Today's date: 2025-06-21
Day of the week: Saturday

2. Birthday Info

Enter your birthday (YYYY-MM-DD):  2006-11-16

You are 18 years old.
Time until your next birthday: 147 days, 1 hours, 30 minutes, 6
seconds

3. Double Day
Double Day is: 2010-01-02

4. N-times Older Day
The day one person is 3 times older is: 2007-07-03
```

```
                                                          # 10.
Project 10

print(" GOUTHAM KB\n1AY24AI035 \nAIML 'M' "
)class Time:
    def __init__(self, hour=0, minute=0, second=0):
        self.total_seconds = hour * 3600 + minute * 60 + second

    def __str__(self):
        h, m, s = self.to_hms()
```

```python
        return f"{h:02}:{m:02}:{s:02}"

    def to_hms(self):
        """Convert total_seconds into hours, minutes, seconds."""
        seconds = self.total_seconds
        hour = seconds // 3600
        seconds %= 3600
        minute = seconds // 60
        second = seconds % 60
        return hour, minute, second

    def time_to_int(self):
        """Return total seconds since midnight."""
        return self.total_seconds

    def increment(self, seconds):
        """Add seconds to time."""
        return Time(0, 0, self.total_seconds + seconds)

    def is_after(self, other):
        return self.total_seconds > other.total_seconds

def int_to_time(seconds):
    return Time(0, 0, seconds)

def main():
    start = Time(9, 45, 0)
    print("Start time:", start)

    duration = Time(1, 35, 0)
    print("Duration:", duration)

    end = start.increment(duration.time_to_int())
    print("End time:", end)

    print("Is end after start?", end.is_after(start))

if __name__ == '__main__':
    main()
```

```
GOUTHAM KB
1AY24AI035
AIML 'M'
Start time: 09:45:00
Duration: 01:35:00
End time: 11:20:00
Is end after start? True
```

```
                                                              # 11.
Project 11

print("Anish B L \n1AY24AI010 \nAIML 'M' ")

class Kangaroo:
    def __init__(self):
        # Initialize with a new empty list (to avoid shared list
issues!)
        self.pouch_contents = []

    def put_in_pouch(self, item):
        self.pouch_contents.append(item)

    def __str__(self):
        content_str = ', '.join(str(item) for item in
self.pouch_contents)
        return f"Kangaroo with pouch contents: [{content_str}]"

kanga = Kangaroo()
roo = Kangaroo()
kanga.put_in_pouch(roo)
print("Kanga:", kanga)
print("Roo:", roo)
```

```
GOUTHAM KB
1AY24AI035
AIML 'M
Kanga: Kangaroo with pouch contents: [Kangaroo with pouch contents:
[]]
Roo: Kangaroo with pouch contents: []
```