ASSIGNMENT-8.3

2303A51528

#TAsk-1 Email Validation using TDD Scenario

Input:

```
import re

def validate_email(email):

    # Regular expression for validating an Email

    regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'


    # If the email matches the regex pattern, it's valid

    if re.match(regex, email):

        return True

    else:

        return False

# # Test cases

# print(validate_email("test@example.com"))  # Should return True

# print(validate_email("invalid.email"))     # Should return False

#user have to enter the input

user_input = input("Enter an email address to validate: ")

print(validate_email(user_input))'''
```

```
PS C:\Users\Jukan> & C:/Users/Jukan/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Jukan/One
Drive/ドキュメント/AI ASSIESSTENT CODING/8.3_ASSMNT_1528.py"
Enter an email address to validate: jukantilohith19@gmail.com
True
PS C:\Users\Jukan> & C:/Users/Jukan/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Jukan/One
Drive/ドキュメント/AI ASSIESSTENT CODING/8.3_ASSMNT_1528.py"
c:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:56: SyntaxWarning: invalid e
scape sequence '\.'
  #regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'  fix this regex for email validation in tas
k 1 and use it in task 3 as well if needed.
Enter an email address to validate: jukantilohith19@gmail.com
True
PS C:\Users\Jukan> jukanti@2005
```

#Task 2 Grade Assignment using Loops Scenario

You are building an automated grading system for an online examination platform.

Requirements

• AI should generate test cases for assign_grade(score) where:

ASSIGNMENT-8.3

2303A51528

90-100 → A

80-89 → B

70-79 → C

60-69 → D

Below 60 → F

Input:

```
def assign_grade(score):
    if 90 <= score <= 100:
        return 'A'
    elif 80 <= score < 90:
        return 'B'
    elif 70 <= score < 80:
        return 'C'
    elif 60 <= score < 70:
        return 'D'
    else:
        return 'F'
#user have to enter the input
score_input = int(input("Enter the score to assign a grade: "))
print(assign_grade(score_input))'''
```

output:

```
PS C:\Users\Jukan> & C:/Users/Jukan/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Jukan/One
Drive/ドキュメント/AI ASSIESSTENT CODING/8.3_ASSMNT_1528.py"
c:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:5: SyntaxWarning: invalid es
cape sequence '\.'
  regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
c:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:56: SyntaxWarning: invalid e
scape sequence '\.'
  #regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'  fix this regex for email validation in tas
k 1 and use it in task 3 as well if needed.
Enter the score to assign a grade: 90
A
```

#Task 3 Sentence Palindrome Checker

ASSIGNMENT-8.3

2303A51528

Scenario

You are developing a text-processing utility to analyze sentences.

Requirements

• AI should generate test cases for is_sentence_palindrome(sentence)

• Ignore case, spaces, and punctuation

• Test both palindromic and non-palindromic sentences

• Example:

– "A man a plan a canal Panama" → True

A raw string is also an option.

#regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$' fix this regex for email validation in task 1 and use it in task 3 as well if needed.

Input:

```
import re

regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'

def is_sentence_palindrome(sentence):

    # Remove spaces, punctuation, and convert to lowercase

    cleaned_sentence = re.sub(r'[^A-Za-z0-9]', '', sentence).lower()


    # Check if the cleaned sentence is equal to its reverse

    return cleaned_sentence == cleaned_sentence[::-1]

#user have to enter the input

sentence_input = input("Enter a sentence to check if it's a palindrome: ")

print(is_sentence_palindrome(sentence_input))'''\
```

output:

```
PS C:\Users\Jukan> & C:/Users/Jukan/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Jukan/On
Drive/ドキュメント/AI ASSIESSTENT CODING/8.3_ASSMNT_1528.py"
c:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:5: SyntaxWarning: invalid e
cape sequence '\.'
  regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
Enter a sentence to check if it's a palindrome: iam very good boy
False
```

ASSIGNMENT-8.3

2303A51528

'''

Task 4: ShoppingCart Class Scenario

You are designing a basic shopping cart module for an e-commerce application.

Requirements

• AI should generate test cases for the ShoppingCart class

• Class must include the following methods:

– add_item(name, price)

– remove_item(name)

– total_cost()

• Validate correct addition, removal, and cost calculation

• Handle empty cart scenarios

Input:

```
class ShoppingCart:
    def __init__(self):
        self.items = {}

    def add_item(self, name, price):
        self.items[name] = price

    def remove_item(self, name):
        if name in self.items:
            del self.items[name]

    def total_cost(self):
        return sum(self.items.values())
#user have to enter the input
cart = ShoppingCart()
while True:
    action = input("Enter action (add/remove/total/exit): ").lower()
    if action == 'add':
```

```
    name = input("Enter item name: ")
    price = float(input("Enter item price: "))
    cart.add_item(name, price)
  elif action == 'remove':
    name = input("Enter item name to remove: ")
    cart.remove_item(name)
  elif action == 'total':
    print(f"Total cost: {cart.total_cost()}")
  elif action == 'exit':
    break
  else:
    print("Invalid action. Please try again.")'''
```

output:

```
c:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:5: SyntaxWarning: invalid es
cape sequence '\.'
  regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
c:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:56: SyntaxWarning: invalid e
scape sequence '\.'
  #regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'  fix this regex for email validation in tas
k 1 and use it in task 3 as well if needed.
Enter action (add/remove/total/exit): add
Enter item name: milk
Enter item price: 30
Enter action (add/remove/total/exit): remove
Enter item name to remove: milk
```

Task 5: Date Format Conversion Scenario

You are creating a utility function to convert date formats for reports.

Requirements

• AI should generate test cases for convert_date_format(date_str)

• Input format must be "YYYY-MM-DD"

• Output format must be "DD-MM-YYYY"

• Example:

– "2023-10-15" → "15-10-2023"

ASSIGNMENT-8.3

2303A51528

Input:

```
def convert_date_format(date_str):
    try:
        year, month, day = date_str.split('-')
        return f"{day}-{month}-{year}"
    except ValueError:
        return "Invalid date format. Please use YYYY-MM-DD."
#user have to enter the input
date_input = input("Enter a date in YYYY-MM-DD format to convert: ")
print(convert_date_format(date_input))
```

output:

```
C:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:5: SyntaxWarning: invalid es
cape sequence '\.'
  regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
C:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING\8.3_ASSMNT_1528.py:56: SyntaxWarning: invalid e
scape sequence '\.'
  #regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'  fix this regex for email validation in tas
k 1 and use it in task 3 as well if needed.
Enter a date in YYYY-MM-DD format to convert: 2005-07-19
19-07-2005
PS C:\Users\Jukan\OneDrive\ドキュメント\AI ASSIESSTENT CODING>
```