# Practical-8

March 14, 2025

## 0.1 Practical 8 :-

**Name:** Sarthak Sanay
**Enrollment No:** 230031101611051

### 0.1.1 Problem Statement 1:-

You are building a word game. Write a function that checks whether two given strings are anagrams (contain the same letters, ignoring spaces and capitalization).
**Test Case 1:-**
*Input:* ("listen", "silent")
*Output:* True
**Test Case 2:-**
*Input:* ("hello", "world")
*Output:* False

[2]:

```
def is_anagram(s1, s2):
    s1 = s1.lower()
    s2 = s2.lower()
    s1 = s1.replace(" ", "")
    s2 = s2.replace(" ", "")
    return sorted(s1) == sorted(s2)

s1 = input("Enter str1: ")
s2 = input("Enter str2: ")

print(is_anagram(s1, s2))
```

```
Enter str1:  listen
Enter str2:  silent

True
```

### 0.1.2 Problem Statement 2:-

You are working on a code formatter. Write a function that converts a given CamelCase string to snake_case (all lowercase with underscores).
**Test Case :-**
*Input:* ("CamelCaseExample")
*Output:* "camel_case_example"

[3]:

```
def code_formatter(camel_str):
    snake_str = camel_str[0].lower()
    for char in camel_str[1: ]:
        if char.isupper():
            snake_str += '_' + char.lower()
        else:
            snake_str += char
    return snake_str

camel_str = input("Enter camel-case string: ")
print(code_formatter(camel_str))
```

```
Enter camel-case string:  CamelCaseExample
```

```
camel_case_example
```

### 0.1.3 Problem Statement 3:-

You are building a user registration system. Write a function that checks whether a given password meets the following criteria: 1. At least 8 characters long 2. Contains at least one uppercase letter 3. Contains at least one lowercase letter 4. Contains at least one digit

[5]:

```
def check_password(password):
    upper, lower, digit = False, False, False
    save = True
    if len(password) < 8:
        print("Password is less than 8 characters long.")
        save = False
    for char in password:
        if char.isupper():
            upper = True
        if char.islower():
            lower = True
        if char.isdigit():
            digit = True
    if upper == False:
        print("Password does not contain any upper-case character.")
        save = False
    if lower == False:
        print("Password does not contain any lower-case character.")
        save = False
    if digit == False:
        print("Password does not contain any digits.")
        save = False

    if save == True:
        print("\nUsername and password set successfully!")
    else:
        print("\nSet another password according to given constraints.")
        password = input("Set password: ")
        check_password(password)

print("User Registration System :-\n")
username = input("Choose username: ")
password = input("Set password: ")
check_password(password)
```

```
User Registration System :-


Choose username:  Administrator
Set password:  pass

Password is less than 8 characters long.
Password does not contain any upper-case character.
Password does not contain any digits.

Set another password according to given constraints.

Set password:  myPassword

Password does not contain any digits.

Set another password according to given constraints.

Set password:  myPass1

Password is less than 8 characters long.

Set another password according to given constraints.

Set password:  MYPASSWORD1

Password does not contain any lower-case character.

Set another password according to given constraints.

Set password:  helloworld1

Password does not contain any upper-case character.

Set another password according to given constraints.

Set password:  myPassword1


Username and password set successfully!
```

### 0.1.4 Problem Statement 4:-

You are working on a text manipulation tool that requires rotating words in a sentence. Implement a Python function `rotate_sentence(sentence, n)` that takes a sentence and an integer n as input and rotates each word in the sentence by n positions to the right.

*Example:* Test case rotating “Hello World” by 1 position results in “World Hello”.

[8]:

```python
def rotate_sentence(sentence, n):
    words = sentence.split()
    if len(words) <= 1:
        return sentence
    n = n % len(words)
    rotated_words = words[-n: ] + words[ :-n]
    return " ".join(rotated_words)

sentence = input("Enter sentence: ")
n = int(input("Enter rotation position: "))
print(rotate_sentence(sentence, n))
```

```
Enter sentence:  My name is Sarthak Sanay
Enter rotation position:  2
```

```
Sarthak Sanay My name is
```

### 0.1.5 Problem Statement 5:-

You are working on a text processing tool and need to determine whether a given string is a palindrome. Write a Python function `is_palindrome(word)` that takes a word as input and returns `True` if it is a palindrome, otherwise `False`.

[11]:

```python
def is_palindrome(word):
    word = word.lower()
    word = word.replace(" ", "")
    rev_word = word[::-1]
    return word == rev_word

word = input("Enter word: ")
print(is_palindrome(word))
```

```
Enter word:  Racecar
```

```
True
```