Practical No 3

Aim: To perform string operations including indexing, slicing, searching, and formatting, along with the use of assert statements.

Theory:

Strings in Python are sequences of characters. Python provides various operations to manipulate and analyze strings. The `assert` statement is used to test assumptions in code and helps in debugging.

String Operations

Indexing

Accessing individual characters using position.

```
text = "Python"
print(text[0])  # Output: P
print(text[-1]) # Output: n
```

Slicing

Extracting parts (substrings) of a string.

```
print(text[1:4])    # Output: yth
print(text[:3])     # Output: Pyt
print(text[::2])    # Output: Pto
```

Searching (in, find, index)

Checking existence or position of a substring.

```
"th" in text            # Output: True
text.find("t")          # Output: 2
text.index("y")         # Output: 1
```

Formatting

Using f-strings or `format()` to format strings dynamically.

```
name = "Akshay"
print(f"Hello, {name}!")
print("Welcome, {}!".format(name))
```

Assert Statement

`assert` checks if a condition is true. If not, it raises an `AssertionError`.

Syntax:

```
assert condition, "Error message"
```

Example:

```
age = 18
assert age >= 18, "User is not an adult"
```

Program

```
# String operations and assertions

message = "Natural Language Processing"

# Indexing
print("First character:", message[0])
print("Last character:", message[-1])
# Slicing
print("First word:", message[:7])
print("Every second character:", message[::2])

# Searching
print("Is 'Language' in message?", "Language" in message)
print("Position of 'P':", message.find('P'))
```

```
# Formatting
name = "Akshay"
print(f"Hello, {name}. Welcome to NLP class!")

# Assertion
assert "Language" in message, "Expected word not found in string"
# assert "Hi" in message, "Expected word not found in string"
```

Output

```
First character: N
Last character: g
First word: Natural
Every second character: NtrllauePoesig
Is 'Language' in message? True
Position of 'P': 16
Hello, Akshay. Welcome to NLP class!
```

1. Take a full name input from the user. Extract the first name and last name using slicing and print a formatted welcome message like:

   "Welcome, [First Name] [Last Name]!"

   ## program:

```
name = input("Enter your full name: ")

 for i in range(len(name)):

    if name[i] == ' ':

        space = i

        break

first = name[:space]

last = name[space + 1:]

print("Welcome,", first, last )
```

output:

python -u "c:\Users\ASUS\OneDrive\Desktop\python practicals\programs\p11.py"

Enter your full name: ROSHAN AWARI

Welcome, ROSHAN AWARI

PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs>

2. Use `assert` to ensure the name is at least 5 characters long.

program:

```
name = input("Enter your full name: ")

assert len(name) >= 5, "Name must be at least 5 characters long"

space = 0

for i in range(len(name)):

    if name[i] == ' ':

        space = i

        break

first = name[:space]

last = name[space + 1:]

print("Welcome,", first, last )
```

output:

PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs> python -u "c:\Users\ASUS\OneDrive\Desktop\python practicals\programs\p11.py"

Enter your full name: ROSH

Traceback (most recent call last):

File "c:\Users\ASUS\OneDrive\Desktop\python practicals\programs\p11.py", line 2, in <module>

assert len(name) >= 5, "Name must be at least 5 characters long"

^^^^^^^^^^^^^^^

AssertionError: Name must be at least 5 characters long