PART-B

1. Demonstrate The Usage Of Basic Regular Expressions

```
import re

text = "Hello BCA Students! Welcome to 4th Sem"

//re.match() tries to match a pattern at the beginning of the string.//
match = re.match("Hello", text)
print("Match:", match.group() if match else None)

//re.search() searches the string for a match, and returns a match object if there is a match.//
search = re.search("BCA", text)
print("Search:", search.group() if search else None)

//re.findall() returns a list containing all matches.//
findall = re.findall(r"[0-9]", text)
print("Findall:", findall)

//re.split() returns a list where the string has been split at each match.//
split = re.split(" ", text) # Splitting by space instead of "ls"
print("Split:", split)

// re.sub() replaces the matches with the text of choice.//
sub = re.sub("4th", "Fourth", text) # Replacing "4th" with "Fourth"
print("Sub:", sub)
```

OUTPUT

```
Match: Hello
Search: BCA
Findall: ['4']
Split: ['Hello', 'BCA', 'Students!', 'Welcome', 'to', '4th', 'Sem']
Sub: Hello BCA Students! Welcome to Fourth Sem
```

2. Demonstarte the Use Of Advanced Regular Expressions for Data Validation

```
import re

def validate_data():
email = input("Enter your email address: ")
phone = input("Enter your phone number: ")
url = input("Enter a URL: ")
password = input("Enter your password: ")

//Simple regular expressions for validation//
email_regex = r'\S+@\S+\.\S+'
```

```
phone_regex = r'\d{10}'
url_regex = r'https?://(www\.)?\S+'
password_regex = r'.{8,}'

//Any character, minimum length of 8//
if not re.fullmatch(email_regex, email):
    print("Invalid Email address")

if not re.fullmatch(phone_regex, phone):
    print("Invalid Phone number")

if not re.fullmatch(url_regex, url):
    print("Invalid URL")

if not re.fullmatch(password_regex, password):
    print("Invalid password. It should be a minimum of 8 characters long.")
```

//Run the function//
validate_data()

OUTPUT

Enter your email address: user@example.com
Enter your phone number: 1234567890
Enter a URL: https://www.example.com
Enter your password: secure123
Invalid Email address
Invalid URL
Invalid password. It should be a minimum of 8 characters long

3. Demonstrate Use Of List

//Define the list//
my_list = [10, 20, 30, 3.142, 'Python', 'BCA']
print("Initial list:", my_list)

//Access list elements by index//
print("Element at Index 0:", my_list[0])
print("Element at Index 2:", my_list[2])

//Change the value of a list item//
my_list[4] = 'Java'
print("List after modifying an item:", my_list)

//Add an item to the end of the list and print//
my_list.append('Skyward')
print("List after appending an item:", my_list)

```
//Remove an item from the list by value and print//
my_list.remove('Java')
print("List after removing an item:", my_list)

//Remove an item from the list by index and print//
del my_list[0]
print("List after deleting an item by index:", my_list)

//Pop an item and print//
print("Popped item:", my_list.pop(1))
print("List after popping an item:", my_list)

//Print index of an item directly//
print("Index of 'BCA':", my_list.index('BCA'))

//Print count of an item directly//
print("Count of 3.142:", my_list.count(3.142))

//Print length of the list directly//
print("Length of the list:", len(my_list))

// Reverse a list and print//
my_list.reverse()
print("Reversed list:", my_list)

//Clear the list and print//
my_list.clear()
print("List after clearing:", my_list)
```

OUTPUT

```
Initial list: [10, 20, 30, 3.142, 'Python', 'BCA']
Element at Index 0: 10
Element at Index 2: 30
List after modifying an item: [10, 20, 30, 3.142, 'Java', 'BCA']
List after appending an item: [10, 20, 30, 3.142, 'Java', 'BCA', 'Skyward']
List after removing an item: [10, 20, 30, 3.142, 'BCA', 'Skyward']
List after deleting an item by index: [20, 30, 3.142, 'BCA', 'Skyward']
Popped item: 30
List after popping an item: [20, 3.142, 'BCA', 'Skyward']
Index of 'BCA': 2
Count of 3.142: 1
Length of the list: 4
Reversed list: ['Skyward', 'BCA', 3.142, 20]
List after clearing: []
```