1.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\K
\sequentialSearch.py"
Enter the size of the list: 5
Enter the elements separated by spaces: 62 87 17 66 59
Enter the target number: 66
66 found at index 3
```

2.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktc
\calculator.py"
Calculator Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter your choice (1-4): 1
Enter the first number: 62
Enter the second number: 87
Result: 149.0
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```

3.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\stringFunctions.py"
Enter a string: saysky2
Length of the string: 7
Uppercase: SAYSKY2
Lowercase: saysky2
Enter a substring to count: sky
Occurrences of the substring: 1
Enter a prefix to check: S
The string does not start with the prefix.
Enter a suffix to check: 2
The string ends with the suffix.
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```

4.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\slectionSort.py"
Enter the size of the list: 5
Enter the elements separated by spaces: 62 87 17 66 59
Original list: [62, 87, 17, 66, 59]
Sorted list: [17, 59, 62, 66, 87]
```

5

- **Output**

```
\stack.py"
Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 1
Enter an item to push onto the stack: 62
Item pushed onto the stack: 62

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 1
Enter an item to push onto the stack: 87
Item pushed onto the stack: 87

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
```

```
Enter your choice (1-7): 3
Stack is not empty.

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 4
Top item: 87

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 17
Invalid choice. Please try again.

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
```

```
Enter your choice (1-7): 6
Stack content: ['62', '87']

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 5
Size of the stack: 2

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 4
Top item: 87

Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
```

```
Menu:
1. Push an item
2. Pop an item
3. Check if the stack is empty
4. Peek at the top item
5. Get the size of the stack
6. Print the entire content of the stack
7. Exit
Enter your choice (1-7): 7
Exiting the program.
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\readAndWrite.py"
Enter the file name: saysky2.txt
File doesn't exist. Do you want to create it? (y/n): y
Enter the data to write to the file: Hello this is a sample text written in saysky2.txt
Data written to the file successfully.
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\readAndWrite.py"
Enter the file name: saysky2.txt
Choose an operation:
1. Read from the file
2. Write to the file
3. Exit
Enter your choice (1, 2, or 3): 1
Data read from the file: Hello this is a sample text written in saysky2.txt
Choose an operation:
1. Read from the file
2. Write to the file
3. Exit
Enter your choice (1, 2, or 3): 2
Enter the data to write to the file: Text has chnaged by user
Data written to the file successfully.
Choose an operation:
1. Read from the file
2. Write to the file
3. Exit
Enter your choice (1, 2, or 3): 1
Data read from the file: Text has chnaged by user
Choose an operation:
1. Read from the file
2. Write to the file
3. Exit
Enter your choice (1, 2, or 3): 3
```

Ln 42, Col 38 Spaces: 4 UTF-8 CRLF {} Python 3.12.3 64-bit Go Live tabnine basic

```
Enter your choice (1, 2, or 3): 3
Exiting the program.
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```

7.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\useBasicRegularExp.py"
Pattern: saysky2
Matches: ['saysky2']

Pattern: \b\w{6}\b
Matches: ['author', 'Python']

Pattern: \b[a-z]{3}\b
Matches: ['the', 'the']

Pattern: [A-Z][a-z]+
Matches: ['The', 'Python']

Pattern: \b[a-zA-Z]+\b
Matches: ['The', 'author', 'of', 'the', 'program', 'is', 'the', 'codes', 'is', 'wrtiten', 'in', 'Python']

Pattern: \b[a-zA-Z]{4,}\b
Matches: ['author', 'program', 'codes', 'wrtiten', 'Python']

Pattern: \b[a-zA-Z]+s\b
Matches: ['is', 'codes', 'is']

Pattern: \b[a-zA-Z]{3}s\b
Matches: []

Pattern: \b[a-zA-Z]{4,}s\b
Matches: ['codes']

PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```

8.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\useOfList.py"
First element: apple
Second element: banana
Last element: orange
Modified list: ['apple', 'grape', 'orange']
Removed item: apple
Updated list: ['grape', 'orange']
Length of the list: 2
The element 'orange' exists in the list.
Elements in the list:
grape
orange
Cleared list: []
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```

9.

- **Output**

```
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC> python -u "c:\Users\10582\Desktop\KMCLU_ACADEMIC\5 semester\Python
\useOfDictionaries.py"
Name: saysky2
Age: 21
City: Ballia
Modified dictionary: {'name': 'saysky2', 'age': 20, 'city': 'Ballia'}
Removed value: Ballia
Updated dictionary: {'name': 'saysky2', 'age': 20}
The key 'name' exists in the dictionary.
Keys: dict_keys(['name', 'age'])
Values: dict_values(['saysky2', 20])
Key-Value Pairs:
name : saysky2
age : 20
Cleared dictionary: {}
PS C:\Users\10582\Desktop\KMCLU_ACADEMIC>
```