SRE TRAINING (DAY 16) - PYTHON PROGRAMMING

---

1) **Remove parentheses program**

remove_parenthesis.py > ...

```python
def remove_parentheses(s):
    stack = []
    result = list(s)  # Convert string to list for easy modification

    # First pass: Remove excess closing brackets ')'
    open_count = 0
    for i, char in enumerate(s):
        if char == '(':
            open_count += 1
        elif char == ')':
            if open_count == 0:  # Unmatched ')'
                result[i] = ''  # Remove it
            else:
                open_count -= 1

    # Second pass: Remove excess opening brackets '(' from the right
    open_count = 0
    for i in range(len(s) - 1, -1, -1):
        if result[i] == '(':
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
root@RheaAlisha:~/python_codes# python3 remove_parenthesis.py
(()())
(())
```

CODE LOGIC

1. **Convert the string into a list for easy modification.**
2. **Traverse left to right and remove unmatched).**
3. **Traverse right to left and remove unmatched (.**
4. **Join the modified list back into a string.**
5. **Return the balanced parentheses string.**

2. Reverse words program

Welcome | remove_parenthesis.py | reverse_words.p

reverse_words.py > ...

```
1   def reverse_sentence(s):
2       return ' '.join(s.split()[::-1])
3
4   print(reverse_sentence("The sky is blue"))
5
6
```

```
root@RheaAlisha:~/python_codes# python3 reverse_words.py
blue is sky The
```

CODE LOGIC

- **Split the sentence into words using `split()`.**
- **Reverse the order of words using slicing `[::-1]`.**
- **Join the reversed words back into a sentence using `' '.join()`.**
- **Return the final reversed sentence without altering individual words.**

3) Anagram program

anagrams.py > ...

```
1   def is_anagram(s1, s2):
2       return sorted(s1) == sorted(s2)
3
4
5   print(is_anagram("listen", "silent"))  # True
6   print(is_anagram("hello", "world"))    # False
7
```

```
root@RheaAlisha:~/python_codes# python3 anagrams.py
True
False
```

CODE LOGIC

- **`sorted(s1)` and `sorted(s2)` convert the strings into sorted lists of characters.**
- **If the sorted versions match, the strings are anagrams.**

Welcome | remove_parenthesis.py | reverse_words.py

anagrams.py > ...

```python
def is_anagram(s1, s2):
    return sorted(s1) == sorted(s2)

print(is_anagram("listen", "silent"))  # True
print(is_anagram("hello", "world"))    # False
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
root@RheaAlisha:~/python_codes# python3 remove_parenthesis.py
(()())
(())

root@RheaAlisha:~/python_codes# python3 reverse_words.py
blue is sky The
root@RheaAlisha:~/python_codes# python3 anagrams.py
True
False
root@RheaAlisha:~/python_codes#
```