PPALMS v0.1 Test Scenario

Authors:

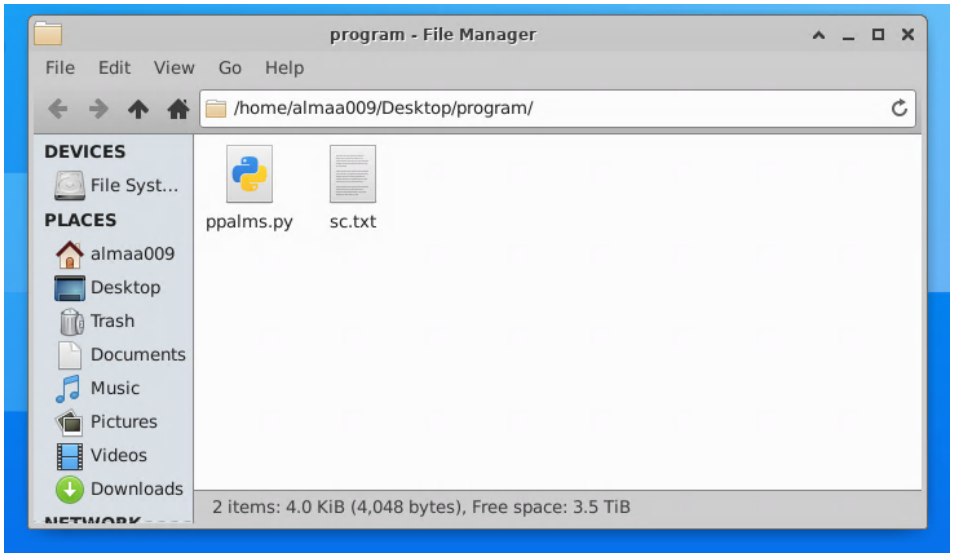
1. Al Yaqdhan Al Maawali
2. Fahia Tabassum
3. Mulki Yusuf
4. Ahmed Al Raisi

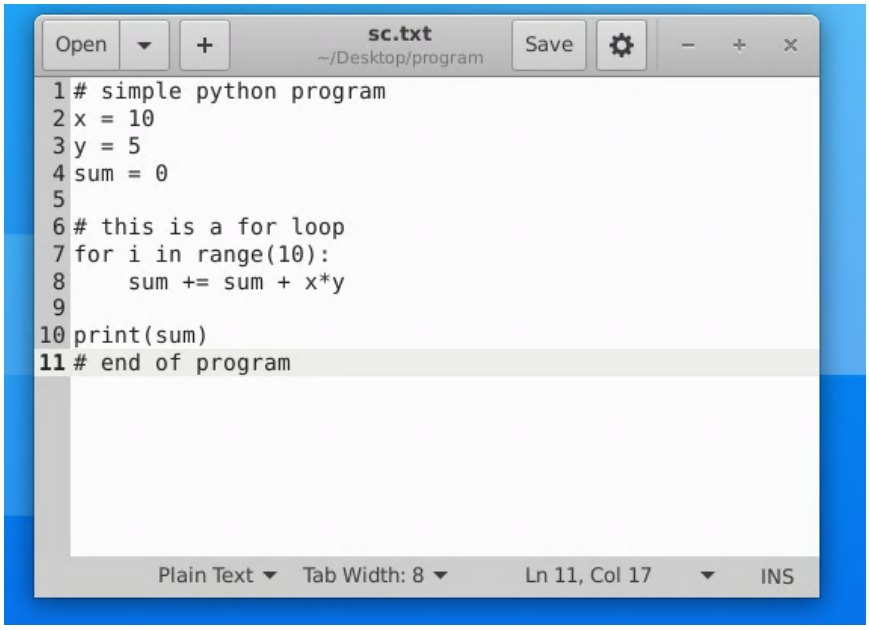
Group: 04

This document shows an end-to-end test scenario of the program. It will include a set-up, execution steps, and how steps satisfy the requirements.

Set-up:

For the set-up, you need to download the ppalms.py file and have python3 downloaded to be able to execute the file. You would also need a text file to act as the source code. Have both of these files in the same folder as so:





Execution steps:

Step 1: Run python file

Run the python file. This could be done using the terminal by typing:

$ python3 ppalms.py

Step 2: Enter file name

Program will ask the user to enter the filename in the terminal. Type the filename, including the filename extension, to the terminal then press enter. If the file does not exist, the program will print an error message and terminate.

```
almaa009@csel-apollo:~/CSCI5801/p4/Csci5801-P4$ python3 ppalms.py
enter filename: sc.txt
```

Requirement Traceability:

This functionality satisfies requirement number 1: import source code of the SRS document. Refer to the Software Requirements Specification document here.

Step 3: Remove single line comments

Program will ask the user if they want to remove single line comments. If yes is typed, the program will remove comments from source code and print the lines. If no or anything else is typed, the program will not remove comments.

```
------------------------
remove comments? (yes/no): yes
------------------------
0 :  x = 10
1 :  y = 5
2 :  sum = 0
3 :  for i in range(10):
4 :      sum += sum + x*y
5 :  print(sum)
------------------------
```

```
------------------------
remove comments? (yes/no): no
------------------------
0 :  # simple python program
1 :  x = 10
2 :  y = 5
3 :  sum = 0
4 :  # this is a for loop
5 :  for i in range(10):
6 :      sum += sum + x*y
7 :  print(sum)
8 :  # end of program
------------------------
```

Step 4: Exclude lines

Program will ask the user if they want to exclude lines. The user should type the value of the lines to be removed in space between the values. For example, 0 2 3 to remove lines 0, 2, and 3. If the user pressed enter without typing anything, the program will not exclude any lines. If the user types lines that do not exist (-1 or 100 for example) the program will ignore those lines and proceed.

```
-----------------------
Enter val of lines to exclude with space between vals or press enter for no exclusion: 0 2 3
-----------------------
0 :  y = 5
1 :      sum += sum + x*y
2 :  print(sum)
-----------------------
```

Requirement Traceability:

This functionality satisfies requirement number 2: annotate lines of the SRS document.

Step 5: Group lines

The program will ask the user if they want to group lines. Lines grouped cannot be rearranged in any order. If no or anything else other than yes is typed, the program will print the lines and terminate. If yes is typed, the program will ask for the value of the start of the line to group and the value of the end of the line. If lines are out of bounds or start of line value is bigger than end of line value, the program will print an error message and the user if they want to group lines again. If no error occurred (start is 1 and end is 2 for example), the program will group those lines, print the lines, and ask the user if they want to group more lines.

```
-----------------------
group lines? (yes/no): yes
-----------------------
enter num of start of line to group: 1
-----------------------
enter num of end of line to group: 2
-----------------------
0 :  y = 5
1.0 :      sum += sum + x*y
1.1 :  print(sum)
-----------------------
group more lines? (yes/no):
```

If the user typed yes, they can group lines again. If the user tries to group lines that are already grouped (line 0 and 1 for example), then an error message will be displayed and the program will ask the user if they want to group lines again. If the user typed no, then lines will be printed and the program terminates.

```
-----------------------
enter num of start of line to group: 0
-----------------------
enter num of end of line to group: 1
-----------------------
--line already in a group--
-----------------------
0 :  y = 5
1.0 :      sum += sum + x*y
1.1 :  print(sum)
-----------------------
group more lines? (yes/no): no
-----------------------
0 :  y = 5
1.0 :      sum += sum + x*y
1.1 :  print(sum)
-----------------------
almaa009@csel-apollo:~/CSCI5801/p4/Csci5801-P4$
```

Requirement Traceability:

This functionality satisfies requirement number 3: form and order line tuples of the SRS document.

This version of the program ends here. The next step of the program should be selecting a question type which would be implemented in future versions of this program.