

LAB MANUAL (Continued.....)

7. RegEx:

- i) Write a Python program to validate email address using regular expression.

Solution:

```
import re

def is_valid_email(email):

    # Regular expression pattern for validating email addresses
    email_pattern = '^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,3})$'

    # Check if the email matches the pattern
    if re.match(email_pattern, email):
        return True
    else:
        return False

# Test the function
email = input("Enter an email address: ")
if is_valid_email(email):
    print("Valid email address.")
else:
    print("Invalid email address.")
```

- ii) Write a Python program to check the validity of a password given by user. Criteria:
 - i. Contain at least 1 letter between A and Z.
 - ii. Contain at least 1 character from \$, *, @, #.
 - iii. Minimum length of password: 8
 - iv. Maximum length of password: 20

Solution:

```
import re
password = "Nam@123Cha$II"
flag = 0
while True:
    if (len(password)<=8 or len(password)>=20):
        flag = -1
        break
    elif not re.search("[a-z]", password):
        flag = -1
        break
    elif not re.search("[A-Z]", password):
        flag = -1
        break
    elif not re.search("[0-9]", password):
        flag = -1
        break
    elif not re.search("[$#@*]", password):
        flag = -1
        break
    elif re.search("\s", password):
        flag = -1
        break
    else:
        flag = 0
        print("Valid Password")
        break

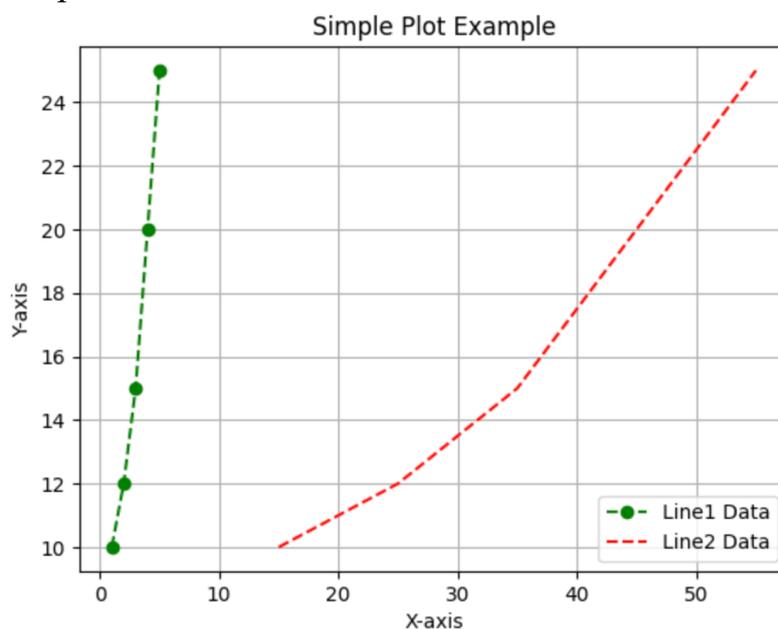
if flag == -1:
    print("Not a Valid Password ")
```


- ii) Write a Python program to plot two lines with different styles, colors, markers, and add a legend, gridlines, and axis labels.

Solution:

```
1 import matplotlib.pyplot as plt
2
3 # Sample data
4 x = [1, 2, 3, 4, 5]
5 y = [10, 12, 15, 20, 25]
6
7 x1 = [15, 25, 35, 45, 55]
8 y1 = [10, 12, 15, 20, 25]
9
10 # Creating a basic plot
11 plt.plot(x, y, linestyle='--', color='g', label='Line1 Data', marker='o')
12 plt.plot(x1, y1, linestyle='--', color='r', label='Line2 Data')
13
14 # Adding a legend
15 plt.legend()
16
17
18 # Adding gridlines
19 plt.grid(True)
20
21 # Adding title and labels
22 plt.title('Simple Plot Example')
23 plt.xlabel('X-axis')
24 plt.ylabel('Y-axis')
25
26 # Displaying the plot
27 plt.show()
```

Output:



10.File Handling:

- i) Write a Python program to read a CSV file containing students details(name, course, marks), filter students with marks above 75%, and save the filtered data into a new file.

Solution:

```
1 import csv
2
3 # Input and output file paths
4 input_file = '/content/sample_data/students.csv'
5 output_file = '/content/sample_data/filtered_students.csv'
6
7 # Read the input CSV file
8 with open(input_file, mode='r') as infile:
9     content = csv.DictReader(infile)
10    filtered_students = []        # Create a list to store filtered student details
11
12    for row in content:
13        if int(row['marks']) > 75:
14            filtered_students.append(row)
15
16 # Write the filtered students to a new CSV file
17 with open(output_file, mode='w', newline='') as outfile:
18     field = content.fieldnames    # Get the fieldnames from the original file
19     writer = csv.DictWriter(outfile, fieldnames=field)
20
21     writer.writeheader()         # Write the header
22     writer.writerows(filtered_students)    # Write the filtered rows
23     print(filtered_students)
24
25 print(f"Filtered data has been saved to {output_file}.")
```

- ii) WAP to add new text to an existing file and print its updated content.

Solution:

```
# Open the file in append mode ('a')
with open('/content/drive/My Drive/foo.txt', 'a') as file:
    # Add new text to the file
    file.write("\nThis is new text added to the file.")

# Open the file in read mode ('r')
with open('/content/drive/My Drive/foo.txt', 'r') as file:
    # Read the updated content of the file
    updated_content = file.read()
    # Print the updated content
print(updated_content)
```

iii) WAP to count total number of lines, words and characters in a file.

```
file_path = "data.txt" # Replace with the path to your file
num_lines, num_chars, num_words = 0, 0, 0
try:
    with open(file_path, 'r') as file:
        print(file.read())

        file.seek(0)
        for line in file:
            #print("line:" , line)
            num_lines += 1
            num_words += len(line.split())
            for word in line.split():
                num_chars= num_chars + len(word)

        print(f"line= {num_lines}, words= {num_words}, Character= {num_chars}")
except FileNotFoundError:
    print(f"File '{file_path}' not found.")
```

PCU/2024/US7044/2024/2024B/NamitaChawla/Sem1/2024B