Q1 a) An email address, i.e. someone@somewhere.com

```
import re
email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b'

def extract_emails_from_file(file_path):
    emails = []
    with open(file_path, 'r') as file:
        text = file.read()
        emails = re.findall(email_pattern, text)
    return emails

file_path = '/content/drive/MyDrive/Colab Notebooks/data.txt'


email_addresses = extract_emails_from_file(file_path)

for email in email_addresses:
    print(email)
```

```
bensul2004nng@spinfinder.com
bensul2004nng@spinfinder.com
200210311310.g9VDANt24674@bloodwork.mr.itd.UM
bensul2004nng@spinfinder.com
obong_715@epatra.com
obong_715@epatra.com
200210312227.g9VMQvDj017948@bluewhale.cs.CU
obong_715@epatra.com
obong_715@epatra.com
webmaster@aclweb.org
obong_715@epatra.com
obong_715@epatra.com
```

ii A URL (http://www.website.com/page or similar)

```
import re

# Define a regular expression pattern to match URLs
url_pattern = r'https?://\S+|www\.\S+'

# Function to extract URLs from a text file
def extract_urls_from_file(file_path):
    urls = []
    with open(file_path, 'r') as file:
        text = file.read()
        urls = re.findall(url_pattern, text)
    return urls

# Provide the path to your text file
file_path = '/content/drive/MyDrive/Colab Notebooks/data.txt'

# Extract URLs from the text file
urls = extract_urls_from_file(file_path)

# Print the found URLs
for url in urls:
    print(url)
```

iii A name listed as lastname, firstname with the first letter of each capitalized

```
import re

# Define a regular expression pattern to match "lastname, firstname" names
name_pattern = r'\b[A-Z][a-zA-Z]*,\s[A-Z][a-zA-Z]*\b'

# Function to extract names from a text file
def extract_names_from_file(file_path):
    names = []
    with open(file_path, 'r') as file:
        text = file.read()
        names = re.findall(name_pattern, text)
    return names

# Provide the path to your text file
file_path = "/content/drive/MyDrive/Colab Notebooks/data.txt"  # Replace with the actual file path

# Extract names from the text file
names = extract_names_from_file(file_path)

# Print the found names
for name in names:
    print(name)
```

```
 Airport, Ikeja
 ELEME, PHD
 However, I
```

iv A phone number, in the format of your choice

```
import re

phone_pattern = r'\b\d{4}\b'

def extract_phone_numbers_from_file(file_path):
    phone_numbers = []
    with open(file_path, 'r') as files:
        text = files.read()
        phone_numbers = re.findall(phone_pattern, text)
    return phone_numbers

file_path = "/content/drive/MyDrive/Colab Notebooks/data.txt"


phone_numbers = extract_phone_numbers_from_file(file_path)

for phone_number in phone_numbers:
    print(phone_number)
```

```
 2002
 2002
 8859
 1998
 1999
 1998
 2002
 2002
 0100
 2919
```

v A data in MM/DD/YY format (be careful about what values are permitted!)

```
import re

date_pattern = r'\b(0[1-9]|1[0-2])/(0[1-9]|1\d|2\d|3[01])/\d{2}\b'


def extract_dates_from_file(file_path):
    dates = []
    with open(file_path, 'r') as files:
        text = files.read()
        dates = re.findall(date_pattern, text)
    return dates

file_path = "/content/drive/MyDrive/Colab Notebooks/data.txt"

dates = extract_dates_from_file(file_path)

for date in dates:
    print(date)
```

Q2-> The files people.txt contain lists of names, email addresses, and phone numbers in a consistent format; open one in a text editor to see what you're dealing with. Your task is to write a regular expression that allows you to selectively extract fields of interest while ignoring extreous information (like white spaces, formatting, etc.).

a. Then, extract the full name and email address of each person and print them in the following format: JohnDoe:username@domain.com

b. Sort the obtained result by first name.

```
import re
pattern = r'([A-Za-z\s]+) <([A-Za-z0-9_.]+@[A-Za-z0-9]+\.[A-Za-z]+)>'

def extract_names_and_emails_from_string(input_string):
    results = []
    with open(input_string, 'r') as files:
      txt = files.read()

      matches = re.findall(pattern, txt)
    for match in matches:
        full_name, email = match
        full_name = ' '.join(full_name.split())

        formatted_name = full_name.replace(' ', '') + ':' + email
        results.append(formatted_name)
    return results


input_string = "/content/drive/MyDrive/Colab Notebooks/peoples.txt"

names_and_emails = extract_names_and_emails_from_string(input_string)


# Print the results
for item in names_and_emails:
    print(item)


print("\nDoing Second Part of the Question------------------------------>\n")


names_and_emails.sort(key=lambda x: x.split(':')[0].lower())
for item in names_and_emails:
  print(item)
```

```
JohnDoe:johndoe@example.com
AliceSmith:alice.smith@example.org
AmanAggarwal:aman@gmail.com

Doing Second Part of the Question------------------------------>

AliceSmith:alice.smith@example.org
AmanAggarwal:aman@gmail.com
JohnDoe:johndoe@example.com
```

PART-> B

You're going to write an interactive calculator! User input is assumed to be a formula that consist of a number, an operator (at least + and -), and another number, separated by white space (e.g. 1 + 1). Split user input using str.split(), and check whether the resulting list is valid: ● If the input does not consist of 3 elements, raise a FormulaError, which is a custom Exception. ● Try to convert the first and third input to a float (like so: float_value = float(str_value)). Catch any ValueError that occurs, and instead raise a FormulaError ● If the second input is not '+' or '-', again raise a FormulaError If the input is valid, perform the calculation and print out the result. The user is then prompted to provide new input, and so on, until the user enters quit.

```
class FormulaError(Exception):
    pass

# Function to perform the calculation
def calculate(first_num, operator, second_num):
    if operator == '+':
        return first_num + second_num

    elif operator == '-':
        return first_num - second_num

    elif operator == '*':
      return first_num * second_num

    else:
        raise FormulaError("Invalid operator")
```

```
while True:
    try:
        user_input = input("Enter a formula (e.g., 1 + 1) or 'quit' to exit: ").strip()

        if user_input.lower() == 'quit':
            break

        elements = user_input.split()

        if len(elements) != 3:
            raise FormulaError("Invalid input format")

        first_num = float(elements[0])
        operator = elements[1]
        second_num = float(elements[2])

        if operator not in ('+', '-'):
            raise FormulaError("Invalid operator")

        result = calculate(first_num, operator, second_num)
        print("Result:", result)

    except ValueError:
        print("ValueError: Invalid number format")
    except FormulaError as e:
        print(f"FormulaError: {e}")
```

```
Enter a formula (e.g., 1 + 1) or 'quit' to exit: 12 + 12
Result: 24.0
Enter a formula (e.g., 1 + 1) or 'quit' to exit: 12 - 12
Result: 0.0
Enter a formula (e.g., 1 + 1) or 'quit' to exit: 122 +- 12
FormulaError: Invalid operator
Enter a formula (e.g., 1 + 1) or 'quit' to exit: 12 // 12
FormulaError: Invalid operator
Enter a formula (e.g., 1 + 1) or 'quit' to exit: quit
```