Question 1- Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).

For the above question : space or any special characters are not allowed in string.

In [1]:
```
import re

def check_character_set(string):
    pattern = r"^[a-zA-Z0-9]+$"
    match = re.match(pattern, string)
    if match:
        return True
    else:
        return False

# Example usage
while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if check_character_set(user_input):
        print("String contains only a-z, A-Z, and 0-9. MATCH \n")
    else:
        print("String contains characters outside a-z, A-Z, and 0-9. NO MAT
```

```
Enter a string (or 'quit' to exit): Speak Now !
String contains characters outside a-z, A-Z, and 0-9. NO MATCH

Enter a string (or 'quit' to exit): Speak Now
String contains characters outside a-z, A-Z, and 0-9. NO MATCH

Enter a string (or 'quit' to exit): SpeakNow
String contains only a-z, A-Z, and 0-9. MATCH

Enter a string (or 'quit' to exit): quit
```

Question 2- Create a function in python that matches a string that has an a followed by zero or more b's

Example of allowed strings are :

1. a
2. ab
3. abbb's
4. ..a..
5. ..ab's..

where .. means any character.

In [2]:

```
def match_a_followed_by_b(string):
    pattern = r'a[b]*'
    return bool(re.search(pattern, string))

# Example usage
while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_a_followed_by_b(user_input):
        print("MATCH found\n")
    else:
        print("NO Match found\n")
```

```
Enter a string (or 'quit' to exit): a
MATCH found

Enter a string (or 'quit' to exit): ab
MATCH found

Enter a string (or 'quit' to exit): abbb
MATCH found

Enter a string (or 'quit' to exit): ccbba
MATCH found

Enter a string (or 'quit' to exit): ccbbabbb
MATCH found

Enter a string (or 'quit' to exit): ccddbbdd
NO Match found

Enter a string (or 'quit' to exit): quit
```

Question 3- Create a function in python that matches a string that has an a followed by one or more b's

Allowed strings are :

1. ab
2. abb's
3. ..ab..

In [3]:

```
import re

def match_a_followed_by_b(string):
    pattern = r'ab+'
    return bool(re.search(pattern, string))

# Example usage

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_a_followed_by_b(user_input):
        print("MATCH found\n")
    else:
        print("NO Match Found\n")
```

```
Enter a string (or 'quit' to exit): a
NO Match Found

Enter a string (or 'quit' to exit): ab
MATCH found

Enter a string (or 'quit' to exit): abbb
MATCH found

Enter a string (or 'quit' to exit): ccbba
NO Match Found

Enter a string (or 'quit' to exit): ccbbabbbb
MATCH found

Enter a string (or 'quit' to exit): ccddbbb
NO Match Found

Enter a string (or 'quit' to exit): quit
```

4. Create a function in Python and use RegEx that matches a string that has an a followed by zero or one 'b'.

Allowed String :

1. a
2. ab
3. ..a..
4. ..ab..

In [4]:

```
import re

def match_string_with_a_but_not_multiple_b(string):
    pattern = r'^.*a(?!b{2,}).*$'
    return bool(re.match(pattern, string))

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_string_with_a_but_not_multiple_b(user_input):
        print("MATCH found\n")
    else:
        print("NO Match Found\n")
```

```
Enter a string (or 'quit' to exit): a
MATCH found

Enter a string (or 'quit' to exit): ab
MATCH found

Enter a string (or 'quit' to exit): ccbba
MATCH found

Enter a string (or 'quit' to exit): ccddbbbabbb
NO Match Found

Enter a string (or 'quit' to exit): ccddabcbb
MATCH found

Enter a string (or 'quit' to exit): ccddbbbb
NO Match Found

Enter a string (or 'quit' to exit): quit
```

Question 5- Write a Python program that matches a string that has an a followed by three 'b'.

Allowed String :

1. abbb
2. ...abbb...

Not allowed

1. ..a..
2. ..ab..
3. ..abbbb..

In [5]:

```
import re

def match_a_followed_by_three_b(string):
    pattern = r'a{1}b{3}(?![b])'
    match = re.search(pattern, string)
    if match:
        return True
    else:
        return False

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_a_followed_by_three_b(user_input):
        print("MATCH Found\n")
    else:
        print("NO Match Found\n")
```

```
Enter a string (or 'quit' to exit): a
NO Match Found

Enter a string (or 'quit' to exit): ab
NO Match Found

Enter a string (or 'quit' to exit): abbbb
NO Match Found

Enter a string (or 'quit' to exit): abbb
MATCH Found

Enter a string (or 'quit' to exit): ccbbbdddabbbnnmm
MATCH Found

Enter a string (or 'quit' to exit): ccddbbabbbb
NO Match Found

Enter a string (or 'quit' to exit): quit
```

Question 6- Write a regular expression in Python to split a string into uppercase letters.

Sample text: “ImportanceOfRegularExpressionsInPython” Output: [‘Importance’, ‘Of’, ‘Regular’, ‘Expression’, ‘In’, ‘Python’]

In [6]:
```
import re

def split_uppercase(string):
    pattern = r'(?<=[a-z])(?=[A-Z])'
    result = re.split(pattern, string)
    return result

# Example usage
input_string = input("Enter a string: ")
uppercase_letters = split_uppercase(input_string)
uppercase_letters = list(filter(None, uppercase_letters))
print(uppercase_letters)
```

```
Enter a string: ImportanceOfRegularExpressionsInPython
['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
```

Question 7- Write a Python program that matches a string that has an a followed by two to three 'b'.

Allowed strings :

1. abb
2. abbb
3. ..abb/bbb..

not allowed

1. ..a..
2. ..ab..
3. ..abbbb's..

In [7]:

```
import re

def match_a_followed_by_two_to_three_b(string):
    pattern = r'a{1}b{2,3}(?![b])'
    match = re.search(pattern, string)
    if match:
        return True
    else:
        return False


# Example usage

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_a_followed_by_two_to_three_b(user_input):
        print("MATCH Found\n")
    else:
        print("NO Match Found\n")
```

```
Enter a string (or 'quit' to exit): a
NO Match Found

Enter a string (or 'quit' to exit): ab
NO Match Found

Enter a string (or 'quit' to exit): abb
MATCH Found

Enter a string (or 'quit' to exit): abbb
MATCH Found

Enter a string (or 'quit' to exit): ccggddhhabbb
MATCH Found

Enter a string (or 'quit' to exit): ccggbbbbbabbbb
NO Match Found

Enter a string (or 'quit' to exit): gghhddbbnnnsss
NO Match Found

Enter a string (or 'quit' to exit): quit
```

Question 8- Write a Python program to find sequences of lowercase letters joined with a underscore.

In [9]:
```
import re

def find_letter_sequences_with_underscore(string):
    pattern = r'[a-zA-Z]+_[a-zA-Z]+'
    result = re.findall(pattern, string)
    return result

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    sequences = find_letter_sequences_with_underscore(user_input)
    print("Sequences found:", sequences)
```

```
Enter a string (or 'quit' to exit): Hi_World_hi_peeps
Sequences found: ['Hi_World', 'hi_peeps']
Enter a string (or 'quit' to exit): quit
```

Question 9- Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'.

Allowed Strings:

1. ..a...b

In [10]:
```
import re

def match_a_followed_by_anything_ending_with_b(string):
    pattern = r'^.*a.*b$'
    return bool(re.match(pattern, string))

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_a_followed_by_anything_ending_with_b(user_input):
        print("MATCH Found\n")
    else:
        print("NO Match Found\n")
```

```
Enter a string (or 'quit' to exit): qqxxyybbbaggghhhb
MATCH Found

Enter a string (or 'quit' to exit): a
NO Match Found

Enter a string (or 'quit' to exit): ab
MATCH Found

Enter a string (or 'quit' to exit): b
NO Match Found

Enter a string (or 'quit' to exit): xxxyyyyaa
NO Match Found

Enter a string (or 'quit' to exit): quit
```

Question 10- Write a Python program that matches a word at the beginning of a string.

In [3]:

```
import re

def match_word_at_beginning(word, string):
    pattern = r'^' + word
    return bool(re.match(pattern, string))

# Example usage

while True:
    input_word = input("Enter a word (or 'quit' to exit): ")

    if input_word.lower() == "quit":
        break

    while True:
        input_string = input("Enter a String (or 'quit' to exit): ")
        if input_string.lower() == "quit":
            break

        if match_word_at_beginning(input_word, input_string):
            print("MATCH ! Word matches at the beginning of the string.")
        else:
            print("NO MAtch ! Word does not match at the beginning of the s
```

```
Enter a word (or 'quit' to exit): Hello
Enter a String (or 'quit' to exit): Hello World
MATCH ! Word matches at the beginning of the string.
Enter a String (or 'quit' to exit): This is fliprobo
NO MAtch ! Word does not match at the beginning of the string.
Enter a String (or 'quit' to exit): quit
Enter a word (or 'quit' to exit): quit
```

Question 11- Write a Python program to match a string that contains only upper and lowercase letters, numbers, and underscores.

In [4]:
```
import re

def match_string_with_allowed_characters(string):
    pattern = r'^(?=.*[A-Z])[a-zA-Z0-9_]+$'
    return bool(re.match(pattern, string))

while True:
    user_input = input("Enter a string (or 'quit' to exit): ")

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

    if match_string_with_allowed_characters(user_input):
        print("MATCH Found\n")
    else:
        print("NO Match Found\n")
```

```
Enter a string (or 'quit' to exit): Aeg3_gh
MATCH Found

Enter a string (or 'quit' to exit): fghtih
NO Match Found

Enter a string (or 'quit' to exit): AFTDG
MATCH Found

Enter a string (or 'quit' to exit): quit
```

Question 12- Write a Python program where a string will start with a specific number.

In [7]:

```
import re

def starts_with_number(number, string):
    pattern = r'^' + re.escape(number)
    return bool(re.match(pattern, string))
while True:
    input_number = input("Enter a number (or 'quit' to exit): ")

    if input_number.lower() == "quit":
        break

    while True:
        input_string = input("Enter a String (or 'quit' to exit): ")
        if input_string.lower() == "quit":
            break

        if starts_with_number(input_number, input_string):
            print("MATCH ! String starts with the specified number.")
        else:
            print("NO MAtch ! String does not start with the specified numb
```

```
Enter a number (or 'quit' to exit): 0
Enter a String (or 'quit' to exit): 0fght
MATCH ! String starts with the specified number.
Enter a String (or 'quit' to exit): ghyiu890
NO MAtch ! String does not start with the specified number.
Enter a String (or 'quit' to exit): 998ggduh
NO MAtch ! String does not start with the specified number.
Enter a String (or 'quit' to exit): quit
Enter a number (or 'quit' to exit): 8
Enter a String (or 'quit' to exit): 87ghdjl
MATCH ! String starts with the specified number.
Enter a String (or 'quit' to exit): quit
Enter a number (or 'quit' to exit): quit
```

Question 13- Write a Python program to remove leading zeros from an IP address

In [1]:
```
def remove_leading_zeros(ip_address):
    # Split the IP address into octets
    octets = ip_address.split(".")

    # Remove leading zeros from each octet
    cleaned_octets = [str(int(octet)) for octet in octets]

    # Join the cleaned octets with dots
    cleaned_ip = ".".join(cleaned_octets)

    return cleaned_ip


# Example usage
ip_address = "192.168.001.001"
cleaned_ip = remove_leading_zeros(ip_address)
print("Cleaned IP address:", cleaned_ip)
```

```
Cleaned IP address: 192.168.1.1
```

Question 14- Write a regular expression in python to match a date string in the form of Month name followed by day number and year stored in a text file.

Sample text : ' On August 15th 1947 that India was declared independent from British colonialism, and the reins of control were handed over to the leaders of the Country'.
Output- August 15th 1947

In [46]:
```
def create_text_file(filename):
    with open(filename, "w") as file:
        while True:
            line = input("Enter text (or 'exit' to quit): ")
            if line.lower() == "exit":
                break
            file.write(line + "\n")
        print(f"Contents have been added to {filename}.")


# Example usage
filename = input("Enter the filename for the text file: ")
create_text_file(filename)
```

```
Enter the filename for the text file: Sample
Enter text (or 'exit' to quit): On August 15th 1947 that India was declared
independent from British colonialism, and the reins of control were handed
over to the leaders of the Country
Enter text (or 'exit' to quit): exit
Contents have been added to Sample.
```

In [47]:
```
import re

def find_dates_in_file(filename):
    with open(filename, "r") as file:
        text = file.read()
        pattern = r"\b([A-Za-z]+ \d{1,2}(?:st|nd|rd|th) \d{4})\b"
        matches = re.findall(pattern, text)
        print(matches)
        return matches

# Example usage
filename = input("Enter the filename of the text file: ")
dates = find_dates_in_file(filename)
print("Date matches found in the file:")
for date in dates:
    print(date)
```

```
Enter the filename of the text file: Sample
['August 15th 1947']
Date matches found in the file:
August 15th 1947
```

Question 15- Write a Python program to search some literals strings in a string. Go to the editor

Sample text : 'The quick brown fox jumps over the lazy dog.' Searched words : 'fox', 'dog', 'horse'

In [25]:
```
def search_words(text, words):
    found_words = []
    for word in words:
        if word in text:
            found_words.append(word)
    return found_words

# Prompt user for input
sample_text = input("Enter the text: ")
searched_words = input("Enter the words to search (separated by spaces): ")

found_words = search_words(sample_text, searched_words)

print("Found words:")
for word in found_words:
    print(word)
```

```
Enter the text: The quick brown fox jumps over the lazy dog
Enter the words to search (separated by spaces): fox dog horse
Found words:
fox
dog
```

Question 16- Write a Python program to search a literals string in a string and also find the location within the original string where the pattern occurs

Sample text : 'The quick brown fox jumps over the lazy dog.' Searched words : 'fox'

In [28]:

```
def search_pattern(text, pattern):
    found_positions = []
    start = 0

    while True:
        position = text.find(pattern, start)
        if position == -1:
            break
        found_positions.append(position)
        start = position + 1

    return found_positions

# Prompt user for input
sample_text = input("Enter the text: ")
searched_word = input("Enter a word to search : ")
found_positions = search_pattern(sample_text, searched_word)

print("Found positions:")
for position in found_positions:
    print(position)
```

```
Enter the text: The quick brown fox jumps over the lazy dog
Enter a word to search : fox
Found positions:
16
```

Question 17- Write a Python program to find the substrings within a string.

Sample text : 'Python exercises, PHP exercises, C# exercises' Pattern : 'exercises'.

In [29]:

```python
import re

def find_substrings(text, pattern):
    substrings = []
    regex = re.compile(pattern)

    matches = regex.finditer(text)
    for match in matches:
        substrings.append(match.group())

    return substrings


# Prompt user for input
sample_text = input("Enter the text: ")
pattern = input("Enter a word to search : ")

found_substrings = find_substrings(sample_text, pattern)

print("Found substrings:")
for substring in found_substrings:
    print(substring)
```

```
Enter the text: 'Python exercises, PHP exercises, C# exercises'
Enter a word to search : exercises
Found substrings:
exercises
exercises
exercises
```

Question 18- Write a Python program to find the occurrence and position of the substrings within a string.

In [30]:

```
import re

def find_substring_occurrences_positions(text, substring):
    occurrences = []
    pattern = re.compile(substring)

    for match in pattern.finditer(text):
        occurrence = {
            'substring': match.group(),
            'start_position': match.start(),
            'end_position': match.end() - 1,
        }
        occurrences.append(occurrence)

    return occurrences

# Example usage

# Prompt user for input
sample_text = input("Enter the text: ")
pattern = input("Enter a word to search : ")

found_occurrences = find_substring_occurrences_positions(sample_text, subst

print("Found occurrences of the substring:")
for occurrence in found_occurrences:
    print(f"Substring: {occurrence['substring']}")
    print(f"Start Position: {occurrence['start_position']}")
    print(f"End Position: {occurrence['end_position']}")
    print()
```

```
Enter the text: Python exercises, PHP exercises, C# exercises
Enter a word to search : exercises
Found occurrences of the substring:
Substring: exercises
Start Position: 7
End Position: 15

Substring: exercises
Start Position: 22
End Position: 30

Substring: exercises
Start Position: 36
End Position: 44
```

Question 19- Write a Python program to convert a date of yyyy-mm-dd format to dd-mm-yyyy format.

In [32]:
```
from datetime import datetime

def convert_date_format(date):
    # Convert the date string to a datetime object
    datetime_obj = datetime.strptime(date, "%Y-%m-%d")

    # Format the datetime object to the desired format
    new_date_format = datetime_obj.strftime("%d-%m-%Y")

    return new_date_format

# Prompt user for input
date = input("Enter a date (yyyy-mm-dd): ")
new_date_format = convert_date_format(date)

print("Original date:", date)
print("Converted date:", new_date_format)
```

```
Enter a date (yyyy-mm-dd): 1998-07-29
Original date: 1998-07-29
Converted date: 29-07-1998
```

Question 20- Write a Python program to find all words starting with 'a' or 'e' in a given string.

In [33]:
```
import re

def find_words_starting_with_a_or_e(text):
    pattern = r"\b[aAeE]\w+\b"
    words = re.findall(pattern, text)
    return words

# Prompt user for input
sample_text = input("Enter the text: ")


found_words = find_words_starting_with_a_or_e(sample_text)

print("Words starting with 'a' or 'e':")
for word in found_words:
    print(word)
```

```
Enter the text: The apple is on the table. An elephant is in the room. Eat
an apple every day.
Words starting with 'a' or 'e':
apple
An
elephant
Eat
an
apple
every
```

Question 21- Write a Python program to separate and print the numbers and their position of a given string.

In [34]:

```python
import re

def separate_numbers_with_positions(text):
    pattern = r"\d+"
    numbers = re.findall(pattern, text)
    positions = [match.start() for match in re.finditer(pattern, text)]

    return numbers, positions

# Prompt user for input
sample_text = input("Enter the text: ")


numbers, positions = separate_numbers_with_positions(sample_text)

print("Numbers and their positions:")
for number, position in zip(numbers, positions):
    print(f"Number: {number} | Position: {position}")
```

```
Enter the text: I have 10 apples, 5 oranges, and 3 bananas.
Numbers and their positions:
Number: 10 | Position: 7
Number: 5 | Position: 18
Number: 3 | Position: 33
```

Question 22- Write a regular expression in python program to extract maximum numeric value from a string

In [35]:
```
import re

def extract_max_numeric_value(text):
    pattern = r"\d+"
    matches = re.findall(pattern, text)

    if matches:
        max_value = max(map(int, matches))
        return max_value
    else:
        return None

# Prompt user for input
sample_text = input("Enter the text: ")


max_value = extract_max_numeric_value(sample_text)

if max_value is not None:
    print("Maximum numeric value:", max_value)
else:
    print("No numeric values found.")
```

```
Enter the text: The maximum value is 42, but there are other numbers like 1
5, 7, and 99
Maximum numeric value: 99
```

Question 23- Write a Regex in Python to put spaces between words starting with capital letters

In [36]:
```
import re

def add_spaces_between_capital_words(text):
    pattern = r"([A-Z][a-z]+)"
    modified_text = re.sub(pattern, r" \1", text)
    return modified_text.strip()


# Prompt user for input
sample_text = input("Enter the text: ")



modified_text = add_spaces_between_capital_words(sample_text)

print("Modified text:", modified_text)
```

```
Enter the text: TheQuickBrownFoxJumpsOverTheLazyDog
Modified text: The Quick Brown Fox Jumps Over The Lazy Dog
```

Question 24- Python regex to find sequences of one upper case letter followed by lower case letters

In [37]:

```
import re

def find_sequences(text):
    pattern = r"[A-Z][a-z]+"
    sequences = re.findall(pattern, text)
    return sequences

# Example usage

# Prompt user for input
sample_text = input("Enter the text: ")



found_sequences = find_sequences(sample_text)

print("Found sequences:")
for sequence in found_sequences:
    print(sequence)
```

```
Enter the text: The quick Brown Fox Jumps over the Lazy dog
Found sequences:
The
Brown
Fox
Jumps
Lazy
```

Question 25- Write a Python program to remove duplicate words from Sentence using Regular Expression

In [38]:
```
import re

def remove_duplicate_words(sentence):
    pattern = r"\b(\w+)\b\s+(?=.*\b\1\b)"
    modified_sentence = re.sub(pattern, "", sentence)
    return modified_sentence.strip()

# Prompt user for input
sample_sentence = input("Enter the text: ")

modified_sentence = remove_duplicate_words(sample_sentence)

print("Modified sentence:", modified_sentence)
```

```
Enter the text: This is is a sample sentence with duplicate words words
Modified sentence: This is a sample sentence with duplicate words
```

Question 26- Write a python program using RegEx to accept string ending with alphanumeric character.

In [40]:
```
import re

def check_string_ending_alphanumeric(string):
    pattern = r"[A-Za-z0-9*&@]$"
    match = re.search(pattern, string)
    if match:
        return True
    else:
        return False

sample_string = input("Enter the text: ")

result = check_string_ending_alphanumeric(sample_string)

if result:
    print("The string ends with an alphanumeric character.")
else:
    print("The string does not end with an alphanumeric character.")
```

```
Enter the text: Hello World @ fliprobo !29
The string ends with an alphanumeric character.
```

Question 27-Write a python program using RegEx to extract the hashtags.

Sample Text: text = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <U+00A0><U+00BD><U+00B1><U+0089> "acquired funds" No wo""" Output: ['#Doltiwal', '#xyzabc', '#Demonetization']

In [41]:
```
import re

def extract_hashtags(text):
    pattern = r"#\w+"
    hashtags = re.findall(pattern, text)
    return hashtags

# Example usage
sample_text = 'RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Der

hashtags = extract_hashtags(sample_text)

print("Extracted hashtags:")
for hashtag in hashtags:
    print(hashtag)
```

```
Extracted hashtags:
#Doltiwal
#xyzabc
#Demonetization
```

Question 28- Write a python program using RegEx to remove <U+..> like symbols

Check the below sample text, there are strange symbols something of the sort <U+..> all over the place. You need to come up with a general Regex expression that will cover all such symbols. Sample Text: "@Jags123456 Bharat band on 28??<U+00A0><U+00BD><U+00B8><U+0082>Those who are protesting #demonetization are all different party leaders" Output: @Jags123456 Bharat band on 28??Those who are protesting #demonetization are all different party leaders

In [42]:
```
import re

def remove_u_symbols(text):
    pattern = r"<U\+[A-Za-z0-9]+>"
    modified_text = re.sub(pattern, "", text)
    return modified_text

# Example usage
sample_text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+0(

modified_text = remove_u_symbols(sample_text)

print("Modified text:", modified_text)
```

```
Modified text: @Jags123456 Bharat band on 28??<ed><ed>Those who are protest
ing #demonetization are all different party leaders
```

Question 29- Write a python program to extract dates from the text stored in the text file.

Sample Text: Ron was born on 12-09-1992 and he was admitted to school 15-12-1999.
Store this sample text in the file and then extract dates.

In [43]:
```
def create_text_file(filename):
    with open(filename, "w") as file:
        while True:
            line = input("Enter text (or 'exit' to quit): ")
            if line.lower() == "exit":
                break
            file.write(line + "\n")
        print(f"Contents have been added to {filename}.")


# Example usage
filename = input("Enter the filename for the text file: ")
create_text_file(filename)
```

```
Enter the filename for the text file: Sample29
Enter text (or 'exit' to quit): Ron was born on 12-09-1992 and he was admit
ted to school 15-12-1999
Enter text (or 'exit' to quit): exit
Contents have been added to Sample29.
```

In [44]:
```
import re

def extract_dates_from_file(file_path):
    dates = []
    pattern = r"\d{2}-\d{2}-\d{4}"

    with open(file_path, 'r') as file:
        text = file.read()
        dates = re.findall(pattern, text)

    return dates

# Example usage
file_path = "Sample29"
dates = extract_dates_from_file(file_path)

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

```
Extracted dates:
12-09-1992
15-12-1999
```

Question 30- Write a Python program to replace all occurrences of a space, comma, or dot with a colon.

Sample Text- 'Python Exercises, PHP exercises.' Output:

Python:Exercises::PHP:exercises:

In [45]:
```
def replace_spaces_commas_dots_with_colons(text):
    replacements = [' ', ',', '.']
    for char in replacements:
        text = text.replace(char, ':')
    return text

# Example usage
sample_text = 'Python Exercises, PHP exercises.'
modified_text = replace_spaces_commas_dots_with_colons(sample_text)

print("Modified text:", modified_text)
```

```
Modified text: Python:Exercises::PHP:exercises:
```

In [ ]:
```

```