endsempractical

December 16, 2025

1 End Semester Practical Exam

Done by: Siddharth Sudhakar (25901335)

1.1 Question 1:

Given A = {1,2,3,4,5}

B = {4,5,6,7}

C = (A|B)-(A&B)

What is the number of elements in set C?

Objective: To find the number of elements in {C} by implementing the equation.

[1]:
```
A = {1,2,3,4,5}
```

[2]:
```
B = {4,5,6,7}
```

[16]:
```
A|B
```

[16]:
```
{1, 2, 3, 4, 5, 6, 7}
```

[17]:
```
A&B
```

[17]:
```
{4, 5}
```

[3]:
```
C = (A|B)-(A&B) ## C is the difference between the union set: A|B
 ↪{1,2,3,4,5,6,7}
                ## and the intersection set: A&B {4,5}
```

[4]:
```
C # Hence the expected output is {1,2,3,6,7}
```

[4]:
```
{1, 2, 3, 6, 7}
```

[22]:
```
print("Number of elements in set C:",len(C))
```

```
Number of elements in set C: 5
```

Answer: The number of elements in set C = 5

1.2 Question 2:

Write a program that simulates mutation on an immutable string without using: - replace() - list() - join()

Replace every k-th occurence of a given character with another character.

Objective: Given a string, replace every k-th occurence of a given character with another character.

[18]:
```
def mutateStr(string, oldChar, newChar, k):
    count = 0
    i = 0
    while i < len(string): # Loop through the string
        if string[i] == oldChar: # Identifies the character to be replaced in␣
↪the string
            count += 1 # Keeps track of count for the specified character
            if count % k == 0: # If the count hits the k-th occurence
                string = string[:i] + newChar + string[i+1:] # Replace the k-th␣
↪occurence with the other specified character
        i += 1
    return string
```

[23]:
```
mutateStr("Sidddharth", 'd', 'f', 2)
```

[23]:
```
'Sidfdharth'
```

Explanation: This mutation algorithm uses string concatenation concept. Here the k-th occurence is replaced by reassigning the updated string to the original string in the required format using string slicing and concatenation.

1.3 Question 3:

Write a program that prints a number, its square and its cube repeatedly in the range (1,n)

Objective: To write a program that given a number n, it prints a number, its square and its cube repeatedly in range (1,n)

[20]:
```
def printNumSqCube(n):
    for i in range(1,n+1): # Loop to traverse in the range (1,n)
        print("The number is:", i) # Print the number i
        print("The number to the power of 2 is:", i**2) # Print the number i's␣
↪square value
        print("The number to the power of 3 is:", i**3) # Print the number i's␣
↪cube value
        print()
```

[21]:
```
printNumSqCube(4)
```

```
The number is: 1
The number to the power of 2 is: 1
```

```
The number to the power of 3 is: 1

The number is: 2
The number to the power of 2 is: 4
The number to the power of 3 is: 8

The number is: 3
The number to the power of 2 is: 9
The number to the power of 3 is: 27

The number is: 4
The number to the power of 2 is: 16
The number to the power of 3 is: 64
```