

PROGRAMMING ASSIGNMENT 1

Course: Skill Based Programming – Python programming

Class: S.E. Computer (Division A and B)

Date of Assignment: 21-02-2024

Date of Submission: 06-03-2024

CSL405.1: Apply basic concepts of python like control statements, in-built data structures, functions and Object Oriented Paradigms.

CSL405.2: Implement file handling techniques and text processing

functionalities. **CSL405.3:** Implement data structures without using libraries.

NOTE: Solve the following questions using Python. Submit code along with appropriate output for all test cases.

Q	Description	CO mapping
Q1	<p>You and Fredrick are good friends. Yesterday, Fredrick received credit cards from ABCD Bank. He wants to verify whether his credit card numbers are valid or not. You happen to be great at regex so he is asking for your help!</p> <p>A valid credit card from ABCD Bank has the following characteristics:</p> <ul style="list-style-type: none">▶ It must start with a 4/5/6.▶ It must contain exactly 16 digits.▶ It must only consist of digits 0 to 9.▶ It may have digits in groups of 4, separated by <i>one</i> hyphen "-". ▶ It must NOT use any other separator like ' ', '_', etc.▶ It must NOT have 4 or more consecutive repeated digits. Test cases: <p>Valid :</p> <p>4578-3456-0978-5069 6234-2221-1234-8900</p> <p>Invalid:</p> <p>1234-4567-3456 5678-7777-1345-3456 4567_5678_3333-4589w</p>	CSL405.2
Q2	<p>Read a file that contains 'n' numbers (represented in binary format). Each number is represented by 16 bits and stored on a new line. The program should sort the number in ascending order and write back result in new file.</p>	CSL405.2

Q3	You are given the pointer to the head node of a sorted linked list, where the data in the nodes is in ascending order. Delete nodes and return a sorted list with each distinct value in the original list. The given head pointer may be null indicating that the list is empty.	CSL405.3
----	---	----------

Q4	Suppose u and v both have values of type set and $u \cap v == u - v$. From this we can conclude that: (Select the correct choice and justify your answer) <ul style="list-style-type: none"> a. u and v are identical b. u and v are disjoint c. u is a subset of v d. v is a subset of u 	CSL405.1
Q5	Suppose u and v both denote sets in Python. Under what condition can we guarantee that $u - (u - v) == v$? (Select the correct choice and justify your answer) <ul style="list-style-type: none"> a. This is true for any u and v. b. The set u should be a subset of v. c. The set v should be a subset of u. d. The sets u and v should be disjoint. 	CSL405.1
Q6	Here is a function to compute the second largest value in a list of distinct positive integers. You have to fill in the missing lines. You can assume that there are at least two numbers in the list. <pre>def secondmax(l): (mymax,mysecondmax) = (0,0) for i in range(len(l)): # Your code below this line # Your code above this line return(mysecondmax)</pre>	CSL405.1
Q7	Here is a recursive function to reverse a list. You have to fill in the missing argument for the recursive call. <pre>def myreverse(l): if l==[]: return(l) else: return(....)</pre>	CSL405.1

RUBRICS for Programming Assignment Grading:

Sr. No	Performance Indicator	Below average	Average	Good	Excellent	Marks
1	On time Submission (2)	-	Submitted after deadline (1)	Early or on time submission(2)		
2	Test cases and output (4)	Incorrect output (1)	Expected output is verified only for few test cases (2)	Expected output is Verified for all test cases but is not presentable (3)	Expected output is obtained for all test cases. Presentable and easy to follow (4)	
3	Coding efficiency (2)	The code is not structured at all.(0)	The code is structured but not efficient (1)	The code is structured and efficient. (2)	-	
4	Knowledge(2)	Basic concepts not clear (0)	Understood the basic concepts (1)	Could explain the concept with suitable example (1.5)	Could relate the theory with real world application(2)	
Total Marks						

Q1

```
import re

def valid_card(card_number):
    pattern = r'^(?!.*(\d)(-?\1){3})[4-6]\d{3}(-?\d{4}){3}$'
    return bool(re.match(pattern, card_number))

# Take input from the user
user_input = input("Enter your credit card number: ")

# Check if the entered credit card number is valid
if valid_card(user_input):
```

```
    print("It is valid")
else:
    print("It is invalid")
```

```
"C:\Users\Mark Lopes\PycharmProjects\Python\.venv\Scripts\python.exe"
Enter your credit card number: 4567-1234-7890-7613
It is valid

Process finished with exit code 0
|
```

Q2

```
def read_numbers_from_file(filename):
    with open(filename, "r") as f:
        lines = f.readlines()
        # Split each line by spaces and convert to integers
        list_of_numbers = [int(number) for line in lines for number in
line.split()]
    return list_of_numbers

def sorted_list(list_of_numbers):
    list_of_numbers.sort()
    return list_of_numbers

def write_numbers_in_file(filename, sorted_numbers):
    with open(filename, "w") as f:
        f.write("Sorted list:\n")
        for number in sorted_numbers:
            f.write(f'{number:016}\n')

def main():
    input_file = "input_file.txt"
    output_file = "output_file.txt"
    numbers = read_numbers_from_file(input_file)
    sorted_numbers = sorted_list(numbers)
    write_numbers_in_file(output_file, sorted_numbers)

    with open(output_file, "r") as f:
        print(f.read())

if __name__ == "__main__":
    main()
```

Input file:

	Q1.py	Q2.py	Q3.py	Q6.py	Q7.py	input_file.txt	output_file.txt
1						0010 0110 0111	
2							

Output file:-

	Q1.py	Q2.py	Q3.py	Q6.py	Q7.py	input_file.txt	output_file.txt
1							Sorted list:
2							0000000000000010
3							0000000000000110
4							0000000000000111
5							

Q3

```
class ListNode:
    def __init__(self, value=0, next=None):
        self.value = value
        self.next = next

def remove_duplicates(head):
    current = head

    while current is not None and current.next is not None:
        if current.value == current.next.value:
            current.next = current.next.next
        else:
            current = current.next

    return head

def create_linked_list():
    values = input("Enter ascended sorted values for the linked list with
space in between: ").split()
    values = [int(value) for value in values]

    if not values:
        return None

    head = ListNode(values[0])
```

```

current = head

for value in values[1:]:
    current.next = ListNode(value)
    current = current.next

return head

def print_linked_list(head):
    current = head
    while current is not None:
        print(current.value, end=" ")
        current = current.next
    print()

# Example usage:
head = create_linked_list()
print("\nOriginal Linked List:")
print_linked_list(head)

if head is not None:
    new_head = remove_duplicates(head)
    print("\nLinked List after Removing Duplicates:")
    print_linked_list(new_head)
else:
    print("Empty linked list.")

```

```

"C:\Users\Mark Lopes\PycharmProjects\Python\.venv\Scripts\python.exe" "C:\Users\Mark Lopes\Pychar
Enter ascended sorted values for the linked list with space in between: 1 2 3 4 4 4 5 5 6 6

Original Linked List:
1 2 3 4 4 4 5 5 6 6

Linked List after Removing Duplicates:
1 2 3 4 5 6

Process finished with exit code 0

```

Q4

$$(u \setminus v = u - v)$$

This implies that every element in (u) is also in (v) , which means (u) is a subset of (v) . Therefore, the correct choice is:

c) u is a subset of v

Q5

For $(u - (u \cap v) = v)$ to hold true, it means that every element in (u) is in (v) , which implies that (u) is a subset of (v) . Therefore, the correct choice is:

b) The set u should be a subset of v .

Q6

```
def secondMax(l):  
    (myMax, mySecondMax) = (0, 0)  
    for i in range(len(l)):  
        if l[i] >= myMax:  
            mySecondMax = myMax  
            myMax = l[i]  
    return (mySecondMax)  
  
l = [2,31,4,2,12,54]  
print(f"Second max element is {secondMax(l)}")
```

```
"C:\Users\Mark Lopes\PycharmProjects\Python  
Second max element is 31  
  
Process finished with exit code 0
```

Q7

```
def myreverse(l):  
    if l == []:  
        return l  
    else:  
        return myreverse(l[1:]) + [l[0]]  
  
l = [1,2,3,4,5]  
print("Reverse is ", myreverse(l))  
print(l[0])
```

```
"C:\Users\Mark Lopes\PycharmProjects\Python\.v  
Reverse is  [5, 4, 3, 2, 1]
```

```
Process finished with exit code 0
```