

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

Department of Computer Engineering

PROGRAMMING ASSIGNMENT 1

Course: Skill Based Programming – Python programming

Class: S.E. Computer (Division B)

Date of Assignment: 15-02-2024

Date of Submission: 29-02-2024

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. <p>Test cases:</p> <p>Valid :</p> <p>4578-3456-0978-5069</p> <p>6234-2221-1234-8900</p> <p>Invalid:</p> <p>1234-4567-3456</p> <p>5678-7777-1345-3456</p> <p>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	<p>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.</p>	CSL405.3

Q4	<p>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)</p> <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	<p>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)</p> <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	<p>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.</p> <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	<p>Here is a recursive function to reverse a list. You have to fill in the missing argument for the recursive call.</p> <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

```
def is_valid(card_no):
    card_no = card_no.replace('-', '')

    if card_no[0] in '456' and card_no.isdigit() and len(card_no) == 16:
        for i in range(len(card_no) - 3):
            if card_no[i] == card_no[i+1] == card_no[i+2] == card_no[i+3]:
                #TO check if all 4 no. r diff
                return "Invalid"
        else:
            return "Valid"
    else:
        return "Invalid"

valid_cards = ["4578-3456-0978-5069", "6234-2221-1234-8900"]
```

```

invalid_cards = ["1234-4567-3456", "5678-7777-1345-3456",
"4567_5678_3333-4589w"]

print("Test cases:")
for card in valid_cards:
    print(f"{card}: {is_valid(card)}")

for card in invalid_cards:
    print(f"{card}: {is_valid(card)}")

```

OUTPUT:

C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe

C:\Users\biswa\PycharmProjects\Exp_1\cc.py

Test cases:

4578-3456-0978-5069: Valid

6234-2221-1234-8900: Valid

1234-4567-3456: Invalid

5678-7777-1345-3456: Invalid

4567_5678_3333-4589w: Invalid

Process finished with exit code 0

Q2

```

# Read numbers from the input file
with open('input.txt', 'r') as input_file:
    numbers = [int(line.strip(), 2) for line in input_file]

# Sort the numbers in ascending order
sorted_numbers = sorted(numbers)

# Write the sorted numbers to a new file
with open('output.txt', 'w') as output_file:
    for number in sorted_numbers:

```

```

        # Convert each number back to binary representation and write to the
file

        binary_representation = format(number, '016b')

        output_file.write(binary_representation + '\n')

# Example usage
print("File Copied Successfully !!!")

# input.txt
# 1101
# 1010
# 1111
# output.txt
# 00000000000001010
# 00000000000001101
# 00000000000001111

```

OUTPUT:

C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe

C:\Users\biswa\PycharmProjects\Exp_1\bn.py

File Copied Successfully !!!

Process finished with exit code 0

Q3

```

class ListNode:

    def __init__(self, val=0, next=None): #That ll bydefault

        self.val = val

        self.next = next

    def delete_duplicates(head):

        if not head:

            return head

        current= head

```

```

        while current.next:
            if current.val == current.next.val:
                current.next = current.next.next
            else:
                current = current.next

        return head

def print_ll(head):
    current = head
    while current:
        print(current.val, end='->')
        current=current.next

    print()

class ListNode:
    def __init__(self, val=0, next=None): #That ll bydefault
        self.val = val
        self.next = next

    def delete_duplicates(head):
        if not head:
            return head

        current= head
        while current.next:
            if current.val == current.next.val:
                current.next = current.next.next
            else:
                current = current.next

        return head

def print_ll(head):
    current = head
    while current:
        print(current.val)
        current=current.next

    print()

```

```
head = ListNode(1)

head.next = ListNode(1)

head.next.next = ListNode(2)

head.next.next.next = ListNode(3)

head.next.next.next.next = ListNode(3)


print("Original Linked List:")
print_ll(head)


new_head = delete_duplicates(head)


print("\nLinked List after Removing Duplicates:")
print_ll(new_head)
```

OUTPUT:

C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe

C:\Users\biswa\PycharmProjects\Exp_1\sortedll.py

Original Linked List:

1

1

2

3

3

Linked List after Removing Duplicates:

1

2

3

Process finished with exit code 0

Q4

If $u \oplus v = u - v$, where \oplus denotes the symmetric difference and $-$ denotes the set difference.

The correct choice is:

a. u and v are identical.

Justification:

$u \oplus v$ is the set of elements that are in either of the sets, but not in both.

$u - v$ is the set of elements that are in u but not in v .

For $u \oplus v$ to be equal to $u - v$, it implies that v has no elements that are not already in u , and vice versa. This means that u and v are identical, and every element in one set is also in the other.

Q5

The expression $u \cap v = v$ represents the intersection of sets u and v being equal to set v .

The correct choice is:

c. The set v should be a subset of u .

Justification:

For the intersection $u \cap v$ to be equal to v , it means that all elements of v are also present in u . In other words, set v should be a subset of set u . If v is not a subset of u , then the intersection $u \cap v$ will include only common elements, and it won't be equal to v .

Q6

```
def secondmax(I):  
    (mymax, mysecondmax) = (0, 0)  
  
    for i in range(len(I)):  
        if I[i] > mymax:  
            mysecondmax = mymax  
            mymax = I[i]  
        elif I[i] > mysecondmax:  
            mysecondmax = I[i]  
  
    return mysecondmax  
  
list= [9,6,0,3,7]  
print(f'List of numbers: {list}')  
print(f'Second largest number is {secondmax(list)}')
```


OUTPUT:

C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe

C:\Users\biswa\PycharmProjects\Exp_1\com1.py

List of numbers: [9, 6, 0, 3, 7]

Second largest number is 7

Process finished with exit code 0

Q7

```
def myreverse(I):  
    if I == []:  
        return []  
    else:  
        return myreverse(I[1:]) + [I[0]]  
list=[1,2,3,4,5,6,7,8,9]  
print(f'Original list: {list}')
```

```
print(f'reverse list: {myreverse(list)}')
```

OUTPUT:

C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe

C:\Users\biswa\PycharmProjects\Exp_1\com2.py

Original list: [1, 2, 3, 4, 5, 6, 7, 8, 9]

reverse list: [9, 8, 7, 6, 5, 4, 3, 2, 1]

Process finished with exit code 0