United International University
Department of Computer Science and Engineering
DS 1501: Programming for Data Science
Assignment : Summer 2024

---

1. (a) Correct the errors in the following Python code snippet and ensure valid variable usage: [3]

```
a = b = 6, 9
if a > b
    a - b == 2_sum
else :
    sum = a + b

print(z"Sum is : {2_sum]")
```

(b) Manually trace the following code segment and find the output of it. [4]

```
hi = 0
hlw = 10
num = [10, 20, 30, 40]
hlw =len(num)
for i in range(hlw,6):
    print(i)
    if hi < len(num) - 1:
        num[hi] = num[hi + 1] - 5
        hi += 1
        print(num)
    hlw -= 2
print(num[-4:-1])
```

2. (a) In the world of **Demon Slayer**, you are tasked with determining if a group of Hashiras can defeat a powerful demon. Write a Python program to take input from the user for the total power of the Hashiras, the number of Hashiras, and the demon's rank. If the user enters the demon's rank as **Upper Moon 1-3**, reduce the Hashiras' power by 30%. If the demon's rank is **Upper Moon 4-6**, reduce the Hashiras' power by 20%. Then, based on the total power of the Hashiras, the number of Hashiras, and **the demon's rank**, determine if they can defeat the demon according to the following rules: [6]

- Upper Moon 1-3: Requires at least 3 Hashiras and a total power of Hashiras ≥ 3000
- Upper Moon 4-6: Requires at least 2 Hashiras and a total power of Hashiras ≥ 2000
- Any other demon: Requires at least 1 Hashira and a total power of Hashiras ≥ 1000

Sample Input	**Sample Output**
Enter the total power of the Hashiras: 4000 Enter the number of Hashiras: 4 Enter the Upper Moon demon's rank (if any): 2	Total Power of Hashiras after reduction: 2800.0 Demon Defeated: No
Enter the total power of the Hashiras: 2500 Enter the number of Hashiras: 2 Enter the Upper Moon demon's rank (if any): 5	Total Power of Hashiras after reduction: 2000.0 Demon Defeated: Yes

Table 1: Sample Input and Output O(Question-2(a))

(b) In the world of ***Jujutsu Kaisen***, you are tasked with determining if a cursed object exists in the special-grade or semi-grade list. Write a Python program that takes a string input from the user and checks if that input string exists in both lists. If the input string exists in both lists, it will print **"Exists in both lists."** If the input string exists in only one of the lists, it will print **"Exists in special-grade list"** or **"Exists in semi-grade list"** accordingly. If the input string does not exist in either list, it will print **"Does not exist in either list."** [4]

3. What will be the output of the following snippet of code? [4]

```
rows = 3
cols = 5
```

Input	Output
Enter a string: Sukuna's Finger	Exists in special-grade list
Enter a string: Black Rope	Exists in both lists
Enter a string: Cursed Womb	Exists in semi-grade list
Enter a string: Cursed Necklace	Does not exist in either list

Table 2: Sample Input and Output of (Question-2(b))

```
for i in range(1, rows + 1):
    for j in range(1, cols + 1):
        if (i + j) % 4 == 0 or (i == 2 and j % 4 == 0):
            print('*', end='')
        else:
            print(' ', end='')
    print()
```

4. What will be the output of the following snippet of code? [4]

```
What will be the output of the following snippet of code? [6]
    n = 3
    for i in range(1, n + 1):
        for j in range(1, (n + i - 1) + 1):
            if j <= n - i:
                print(" ", end = "")
            else:
                print("*", end = "")
        print()
    for i in range(n - 1, 0, -1):
        for j in range(1, (n + i - 1) + 1):
            if j <= n - i:
                print(" ", end = "")
            else:
                print("*", end = "")
            print()
```

5. Write a Python program to take input from the user for the total cost, weight of the items, and discount code (if any). If the user enters the discount code "SAVE10", apply a 10% discount to the total cost. Then, based on the total cost and weight of the items, calculate the delivery charge according to the following rules: [5]

Total Cost	Weight Range	Delivery Charge
Less than Tk. 50	Less than or equal to 5 kg	Tk. 5
Less than Tk. 50	Greater than 5 kg	Tk. 15
Tk. 50 - Tk. 100	Less than or equal to 5 kg	Tk. 20
Tk. 50 - Tk. 100	Greater than 5 kg	Tk. 25
Greater than Tk. 100	Any	Tk. 30

Print the total cost and delivery charge. Table 3 shows the sample input and output of the program.

6. Show the manual tracing table for the following code.

```
numbers = [10, 20, 30, 40, 50]
i = 0
```

Sample Input	**Sample Output**
Enter the total cost: Tk. 70 Enter the weight of the items in kg: 4 Enter the discount code (if any): SAVE10	Total cost: Tk. 63.00 Delivery charge: Tk. 20

Table 3: Sample Input and Output (Question-2(a))

.

```
while i < len(numbers) - 1:

     temp = numbers[i]
     numbers[i] = numbers[i + 1]
     numbers[i + 1] = temp
     i += 2

 print("List after swapping:", numbers)
```