Repetition (Looping)

Discussion 1

1 Steps to Solve the Problem

1. **Input Data**:
   - A list or array containing the daily temperature values for the entire month. Assume the month has 30 days for simplicity.
2. **Iterate Through Data**:
   - Loop through each temperature value in the list.
3. **Condition Check**:
   - For each temperature value, check if it exceeds 25 degrees Celsius.
4. **Count Days**:
   - Maintain a counter to keep track of the number of days where the temperature exceeds 25 degrees.
5. **Output the Result**:
   - Print or return the count of days with temperatures above 25 degrees.

```
##Hardcode

# List of daily temperatures for one month (30 days)

daily_temperatures = [22, 25, 27, 23, 28, 26, 24, 30, 31, 29, 21, 22, 26, 27, 28, 29, 25, 30, 31, 32, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]


# Initialize a counter for days with temperatures above 25 degrees

days_above_25 = 0


# Loop through each day's temperature

for temperature in daily_temperatures:

    # Check if the temperature is above 25 degrees

    if temperature > 25:

        days_above_25 += 1


# Output the result
```

```
print("Number of days with temperatures above 25 degrees:", days_above_25)


##User input

# Initialize an empty list to store the daily temperatures

daily_temperatures = []


# Get user input for daily temperatures

print("Please enter the daily temperatures for one month (30 days):")

for i in range(30):

    temperature = float(input(f"Day {i+1}: "))

    daily_temperatures.append(temperature)


# Initialize a counter for days with temperatures above 25 degrees

days_above_25 = 0


# Loop through each day's temperature

for temperature in daily_temperatures:

    # Check if the temperature is above 25 degrees

    if temperature > 25:

        days_above_25 += 1


# Output the result

print("Number of days with temperatures above 25 degrees:", days_above_25)
```

Discussion 2

Answer	Discussion
first 6	`value % 2 == 0 is True, enter True, although value % 3 == 0 is True, it is skipped`
third 7	6 + 1 =7, display “third 7” **pass** has no effect (does nothing) but helps in **indicating** an empty statement/ suite/ block. When 7 + 1, value becomes 8, • The **continue** statement continues with the next iteration of the loop. • Skip some portion of the **while** suite we are executing and have control flow back to the beginning of the **while** loop. • Exit early from this iteration of the loop (not the loop itself), and keep executing the **while** loop.
third 9	
third 10	
fourth 10	While-else • It is entered after the **while** loop’s Boolean expression becomes False. • This entry occurs even when the expression is initially False and the **while** loop has never run. • A handy way to perform some final tasks when the loop ends normally.
fifth 10	Statement after while loop

Repetition (Looping)

Discussion 3

errors

- variable count has not been initialized
- Wrong use of break, results in all letter not equal to ‘a’ contribute to count
- Condition update is missing, looping never end. Sentinel condition is missing.
- Wrong Indentation of (print ()) causes output of every input, instead of being summary of result analysis
- Not error, but need avoid: str is a built in function.

Code with errors:

```
while True:
    str = input("enter a string: ")
    for letter in str:
      if letter == 'a':
          break
      count +=1

    print(count , "strings with letter 'a'")
```

Suggested code (two versions)

```
count=0
while True:
    str_sentinal  = input("enter a string (enter #### to stop): ")
    if str_sentinal =="####":
        break
    for letter in str_sentinal :
      if letter == 'a':
          count +=1
          break


print(count , "strings with letter 'a'")
```

```
count = 0
str_sentinal = input("enter a string (enter #### to stop): ")
while str_sentinal != "####":
    for letter in str_sentinal:
      if letter == 'a':
          count +=1
          break
    str_sentinal = input("enter a string (enter #### to stop): ")
print(count , "strings with letter 'a'")
```

Discussion 4

Suggested Solution:

```
for num in range(1, 21):
    if num % 15 ==0:
        print("FizzBuzz")
    elif num %3 ==0:
        print("Fizz")
    elif num %5==0:
        print("Buzz")
    else:
        print(num)
```

Discussion 5

Suggested Solution:

Using two nested for loops: one for upper half and the other for lower half

```
width = int(input("Please enter pattern width: "))

for i in range(1, width+1):
    for j in range(i):
        print("*",end="")
    print()
for i in range(width-1,0, -1):
    for j in range(i):
        print("*",end="")
    print()
```

Repetition (Looping)

```
max = input ("Please enter an integer for the maximum width of the pyramid: ")
max_int = int(max)
for count in range(1, max_int+1):
    print(count * "*")
for count in range(max_int, 0, -1):
    print(count * "*")
```

Using just two for loops: outer for goes through each row

```
width = int(input("Please enter pattern width: "))


for i in range (1,width * 2):
    if i < width:
        count = i
    else:
        count = width * 2 - i
    for j in range (count):
        print("*", end ="")
    print()
```

Using only one for loop

```
width=int(input('Please enter pattern width: '))
for i in range (1,width * 2):
    if i < width:
        count = i
    else:
        count = width * 2 - i
    print("*"*count)
```