Problem-01

```
def thing():
  print("watch")
  print("process")

thing()
print("gap")
thing()
```

```
watch
process
gap
watch
process
```

problem-02

```
x = 7
print("give me the text:")
def text():
  print("This is a print statement under the defiined function")

text()
print("now do the calculation:")
x = x + 2
print(x)
```

```
give me the text:
anybody can dance
now do the calculation:
9
```

return values

```
def great():
  return "hello"

print(great(), "jenny")
print(great(), "edward")
```

```
hello jenny
hello edward
```

Parameter

```
def language(temp):
  if temp == 'Spanish':
    print("Hola")
  elif temp == 'French':
    print("Bonjour")
  else:
    print("hello")

language("Spanish")
```

```
Hola
```

```
def is_valid_sequence(sequence):
    """Check if the DNA sequence is valid (contains only A, C, G, T)."""
    for nucleotide in sequence:
        if nucleotide not in "ACGT":
            return False
    return True

def calculate_complementary_sequence(sequence):
    """Calculate the complementary DNA sequence."""
    complementary_sequence = ""
    for nucleotide in sequence:
        if nucleotide == 'A':
            complementary_sequence += 'T'
        elif nucleotide == 'T':
            complementary_sequence += 'A'
        elif nucleotide == 'C':
            complementary_sequence += 'G'
        elif nucleotide == 'G':
            complementary_sequence += 'C'
    return complementary_sequence

def calculate_gc_content(sequence):
    """Calculate the GC content of the DNA sequence."""
    gc_count = sequence.count('G') + sequence.count('C')
    return (gc_count / len(sequence)) * 100

while True:
    dna_sequence = input("> ").upper()

    # Exit condition
    if dna_sequence == "EXIT":
        break

    # Validate the DNA sequence
    if not is_valid_sequence(dna_sequence):
        print("Invalid DNA sequence.")
        continue

    # Calculate the complementary sequence and GC content
    complementary_sequence = calculate_complementary_sequence(dna_sequence)
    gc_content = calculate_gc_content(dna_sequence)
```

Start coding or generate with AI.

```
    print("GC Content: {:.2f}%".format(gc_content))
```

```
> aacc
Complementary Sequence: TTGG
GC Content: 50.00%
> acga
Complementary Sequence: TGCT
GC Content: 50.00%
> atcc
Complementary Sequence: TAGG
GC Content: 50.00%
> aaat
Complementary Sequence: TTTA
GC Content: 0.00%
> exit
```