

AI Assisted Coding – Lab Report

Name: Mohammad Muneer Ahmed
Roll No: 2303A51475

Task 1 — Palindrome Number Check

Algorithm:

- 1 Read integer
- 2 Convert to string
- 3 Reverse string
- 4 Compare
- 5 Print result

Pseudocode:

```
input n
s = string(n)
if s == reverse(s)
print Palindrome
else print Not
```

```
↳ day5.py > ...
1  def is_palindrome_number(n):
2      s = str(n)
3      return s == s[::-1]
4
5
6  num = int(input("Enter number: "))
7  if is_palindrome_number(num):
8      print("Palindrome")
9  else:
10     print("Not Palindrome")
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Enter number: 23
Not Palindrome
mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Task 2 — Factorial

Algorithm:

- 1 Read n
- 2 If n<0 invalid
- 3 result=1
- 4 Loop multiply
- 5 Print

Pseudocode:

```
input n
if n<0 stop
result=1
for i 1..n result*=i
print result
```

The screenshot shows a code editor with a dark theme. On the left, there is a file named 'day5.py' containing the following Python code:

```
day5.py > ...
1  def factorial(n):
2      if n < 0:
3          return None
4      result = 1
5      for i in range(1, n + 1):
6          result *= i
7      return result
8
9
10 num = int(input("Enter number: "))
11 ans = factorial(num)
12
13 if ans is None:
14     print("Invalid input")
15 else:
16     print("Factorial =", ans)
17
```

Below the code editor, there is a terminal window showing the execution of the script:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
● mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Enter number: 12
Factorial = 479001600
◊ mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Task 3 — Armstrong

Algorithm:

- 1 Read n
- 2 Count digits
- 3 Sum powers
- 4 Compare
- 5 Print

Pseudocode:

```
input n
p=len(digits)
total=sum(d^p)
if total==n Armstrong
```

The screenshot shows a code editor with Python code for checking Armstrong numbers and a terminal window showing the execution and output.

Code Editor Content:

```
day5.py > ...
1 def is_armstrong(n):
2     if n < 0:
3         return False
4     s = str(n)
5     power = len(s)
6     total = sum(int(d) ** power for d in s)
7     return total == n
8
9
10 num = int(input("Enter number: "))
11 if is_armstrong(num):
12     print("Armstrong Number")
13 else:
14     print("Not an Armstrong Number")
15
```

Terminal Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
● mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Enter number: 11
Not an Armstrong Number
◇ mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Task 4 — Prime or Composite

Algorithm:

- 1 Read n
- 2 Check ≤ 1
- 3 Check even
- 4 Test odd divisors
- 5 Print

Pseudocode:

```
input n
if <=1 neither
test divisors
if divides composite
else prime
```

```
day5.py > ...
1  def classify_number(n):
2      if n <= 1:
3          return "Neither prime nor composite"
4
5      if n == 2:
6          return "Prime"
7
8      if n % 2 == 0:
9          return "Composite"
10
11     i = 3
12     while i * i <= n:
13         if n % i == 0:
14             return "Composite"
15         i += 2
16
17     return "Prime"
18
19
20 num = int(input("Enter number: "))
21 print(classify_number(num))
22

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
● mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Enter number: 14
Composite
◆ mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```

Task 5 — Perfect Number

Algorithm:

- 1 Read n
- 2 Sum divisors
- 3 Use sqrt loop
- 4 Compare
- 5 Print

Pseudocode:

```
input n
sum=1
add divisor pairs
if sum==n perfect
```

```
⚡ day5.py > ...
1  def is_perfect(n):
2      if n <= 1:
3          return False
4
5      total = 1
6      for i in range(2, int(n ** 0.5) + 1):
7          if n % i == 0:
8              total += i
9          if i != n // i:
10             total += n // i
11
12  return total == n
13
14
15 num = int(input("Enter number: "))
16 if is_perfect(num):
17     print("Perfect Number")
18 else:
19     print("Not Perfect Number")
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
● mohammadmuneerahmed@Muneers-MacBook-Air training2.py % python -u "/Users/mohammadmuneerahmed/Documents/training2.py/day5.py"
Enter number: 15
Not Perfect Number
↳ mohammadmuneerahmed@Muneers-MacBook-Air training2.py %
```