

Python Developer Task - 1 Report

Developer Name: Debobrat Borah

Submission Date: 18-7-25

1. Overview

This report contains the solutions and explanations for the Python Developer Task - 1 assigned by Main Flow Services and Technologies Pvt. Ltd. The tasks include basic Python problems along with a custom-designed encryption-decryption system. The project emphasizes algorithm design, coding skills, and documentation.

2. Task Breakdown with Source Codes

2.1 Sum of Two Numbers

```
def sum_of_two_numbers(a, b):  
    return a + b
```

2.2 Odd or Even

```
def check_odd_even(number):  
    return "Even" if number % 2 == 0 else "Odd"
```

2.3 Factorial Calculation

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
    return result
```

2.4 Fibonacci Sequence

```
def fibonacci_sequence(n):  
    sequence = []
```

```
a, b = 0, 1
for _ in range(n):
    sequence.append(a)
    a, b = b, a + b
return sequence
```

2.5 Reverse a String

```
def reverse_string(s):
    return s[::-1]
```

2.6 Palindrome Check

```
def is_palindrome(s):
    return s == s[::-1]
```

2.7 Leap Year Check

```
def is_leap_year(year):
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
```

2.8 Armstrong Number Check

```
def is_armstrong(number):
    digits = str(number)
    power = len(digits)
    total = sum(int(digit) ** power for digit in digits)
    return total == number
```

2.9 Custom Encryption-Decryption System (Caesar Cipher)

```
def encrypt(message, shift):
    encrypted = "
```

```

for char in message:
    if char.isalpha():
        offset = 65 if char.isupper() else 97
        encrypted += chr((ord(char) - offset + shift) % 26 + offset)
    else:
        encrypted += char
return encrypted

```

```

def decrypt(encrypted_message, shift):
    return encrypt(encrypted_message, -shift)

```

3. Key Challenges and Solutions

Task	Challenge	Solution
Factorial	Handling n = 0	Returned 1 for n = 0
Fibonacci	Handling n = 0 or n = 1 cases	Added edge case checks
Armstrong	Correct digit power calculation	Used length of digits in power calc
Encryption	Handling special characters & wrap	Checked char type and used ASCII math

4. Learning Outcomes

- Mastered Python basics: loops, conditions, functions.
 - Understood algorithmic logic and problem-solving.
 - Learned how simple encryption algorithms work.
 - Developed time management skills by adhering to deadlines.
 - Improved documentation skills for professional reporting.
-

5. Conclusion

Completing this task helped me strengthen my Python programming fundamentals and develop a deeper understanding of algorithm design and basic security principles.

6. Attachments

- Python Source Codes for all tasks (included in this report)
- Sample outputs (available on request)
- This report (PDF format)