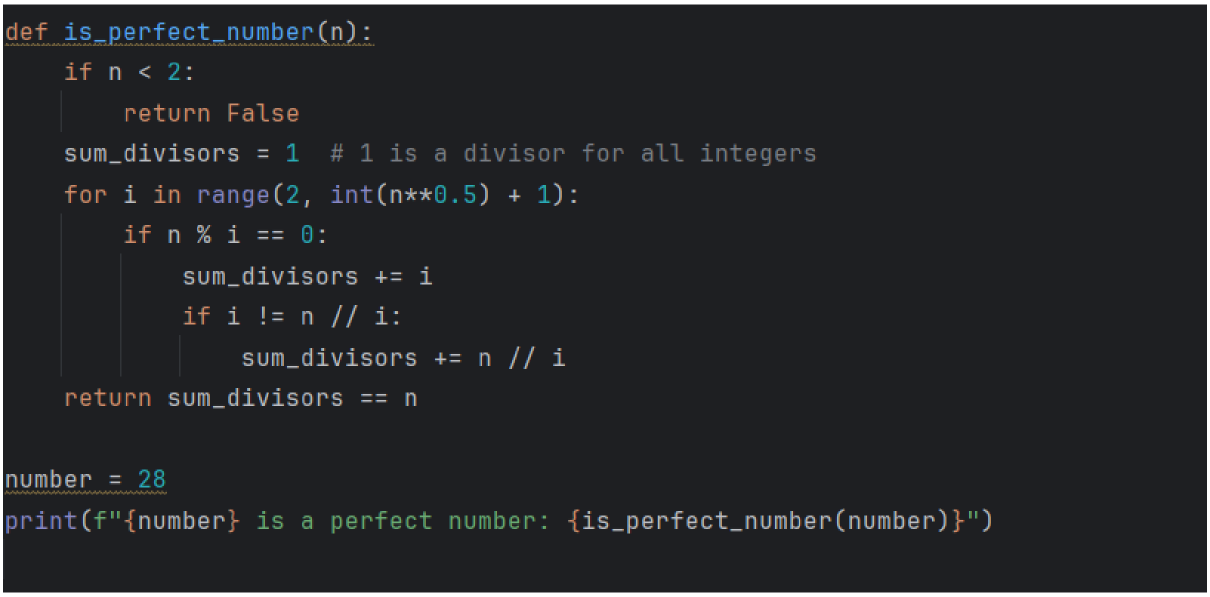
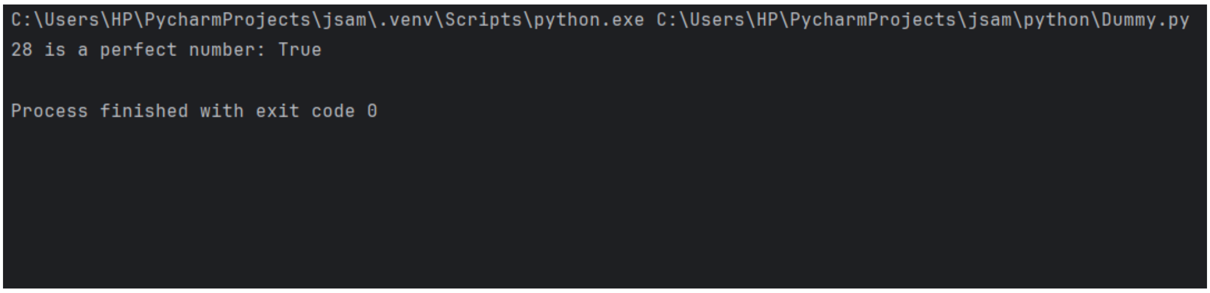
ASSIGNMENT – 4

1.Write a program to find the perfect number.



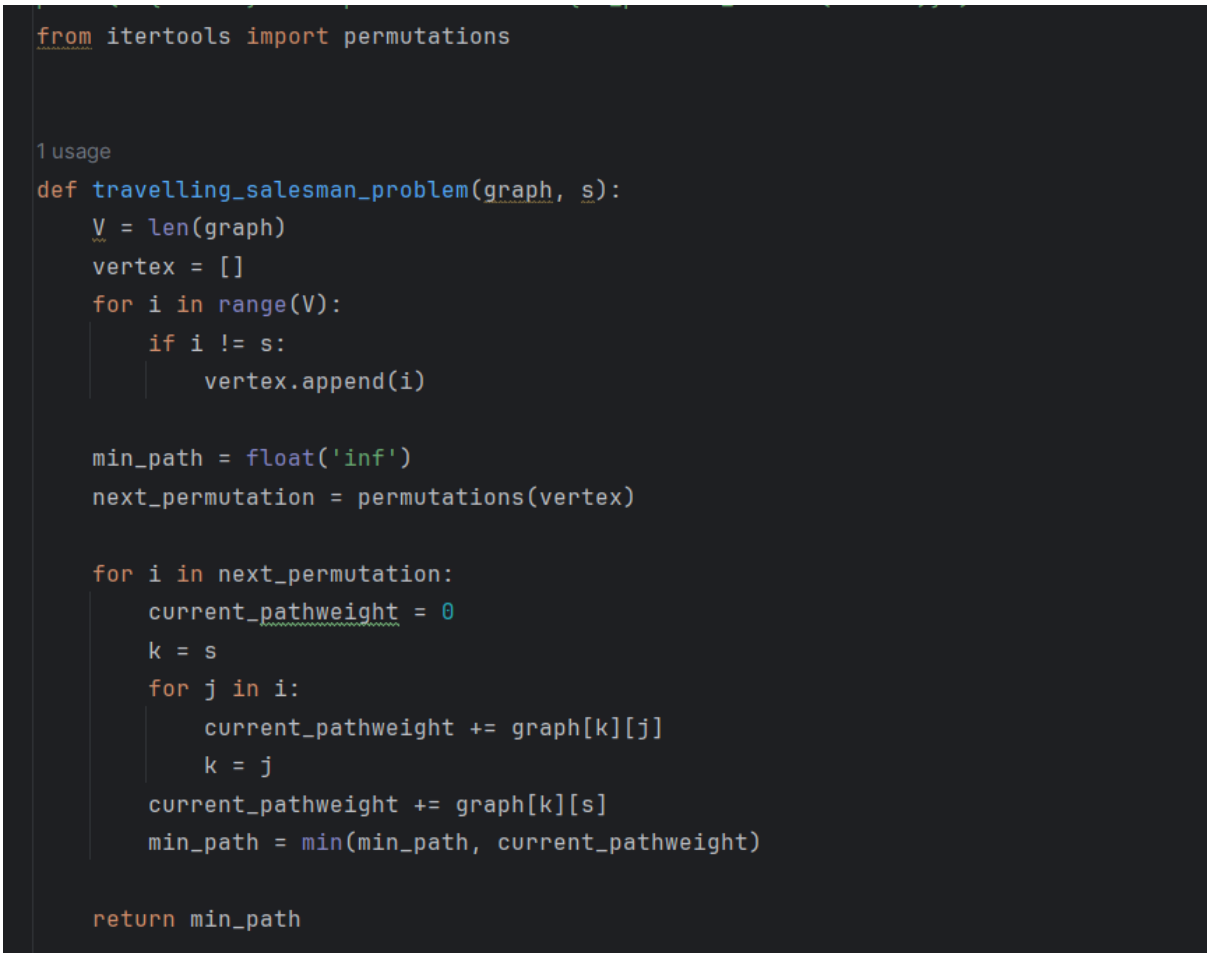
```python
def is_perfect_number(n):
    if n < 2:
        return False
    sum_divisors = 1  # 1 is a divisor for all integers
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            sum_divisors += i
            if i != n // i:
                sum_divisors += n // i
    return sum_divisors == n

number = 28
print(f"{number} is a perfect number: {is_perfect_number(number)}")
```

OUTPUT:



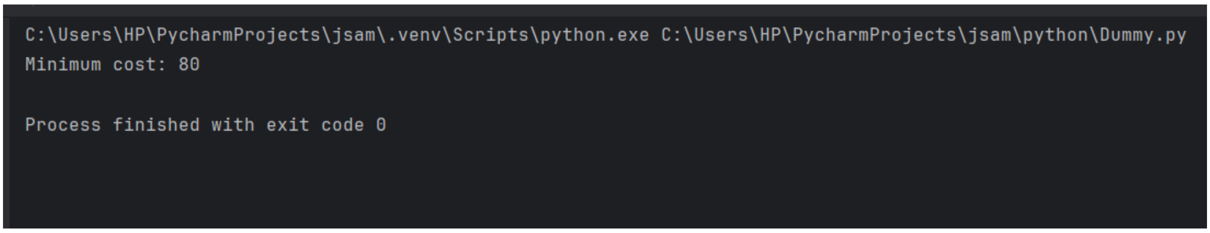
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
28 is a perfect number: True

Process finished with exit code 0
```

2. Program to Perform Travelling Salesman Problem using Dynamic Programming



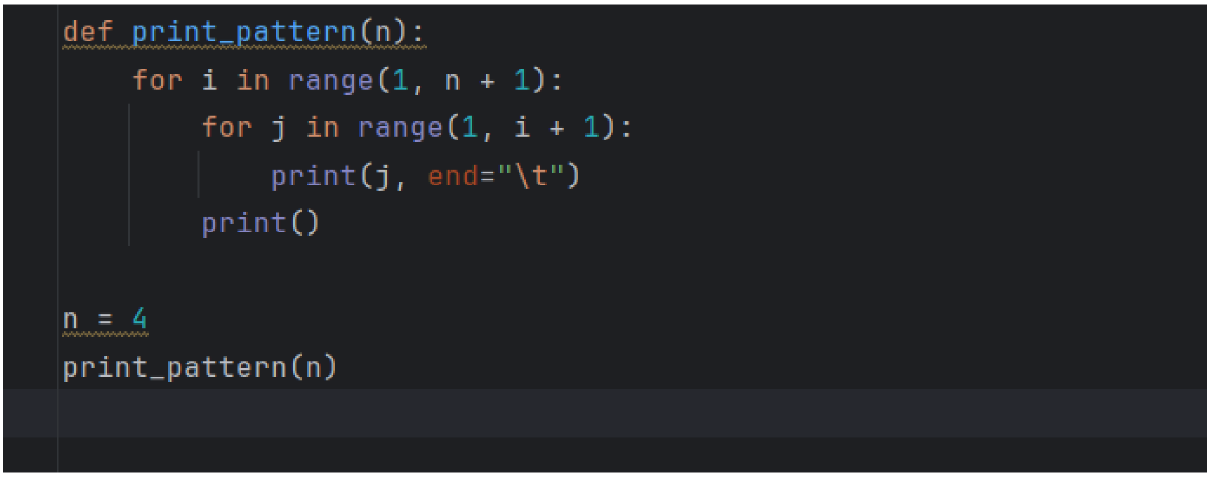
```
from itertools import permutations


def travelling_salesman_problem(graph, s):
    V = len(graph)
    vertex = []
    for i in range(V):
        if i != s:
            vertex.append(i)

    min_path = float('inf')
    next_permutation = permutations(vertex)

    for i in next_permutation:
        current_pathweight = 0
        k = s
        for j in i:
            current_pathweight += graph[k][j]
            k = j
        current_pathweight += graph[k][s]
        min_path = min(min_path, current_pathweight)

    return min_path
```

OUTPUT:



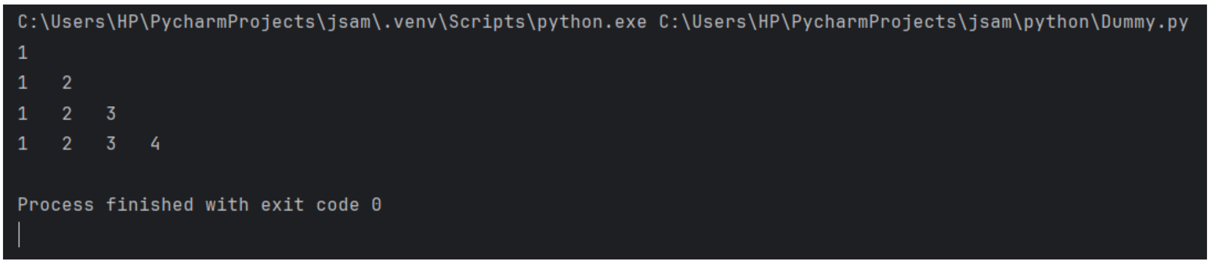
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
Minimum cost: 80

Process finished with exit code 0
```

3. Program for the Given Pattern



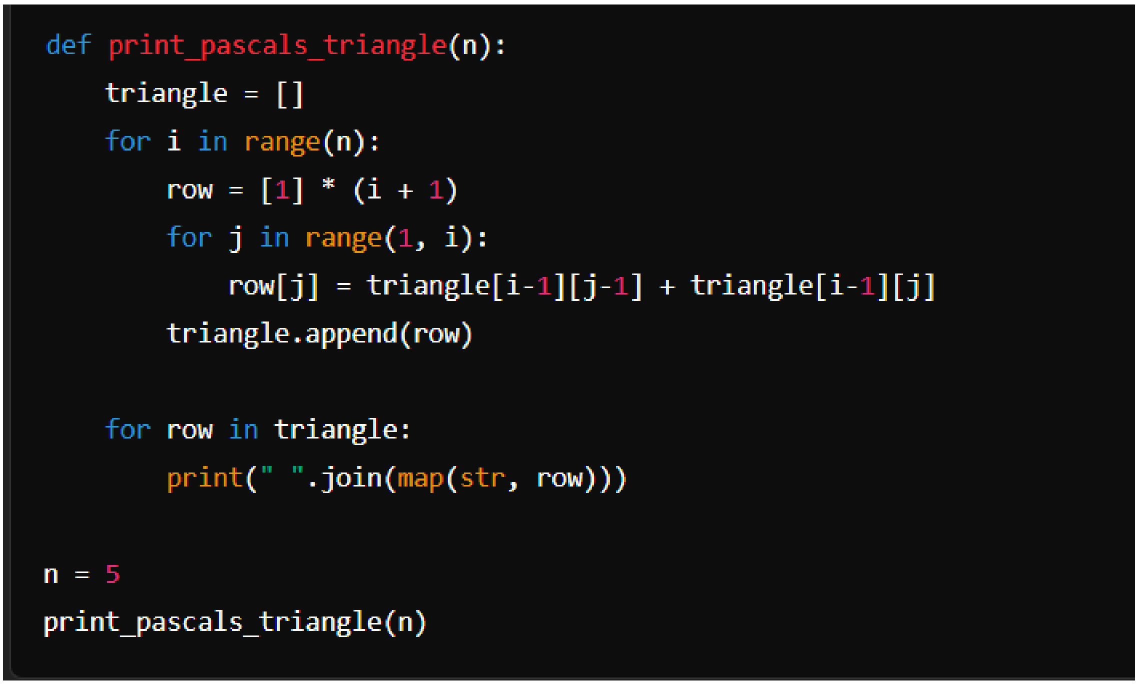
```python
def print_pattern(n):
    for i in range(1, n + 1):
        for j in range(1, i + 1):
            print(j, end="\t")
        print()

n = 4
print_pattern(n)
```

OUTPUT:



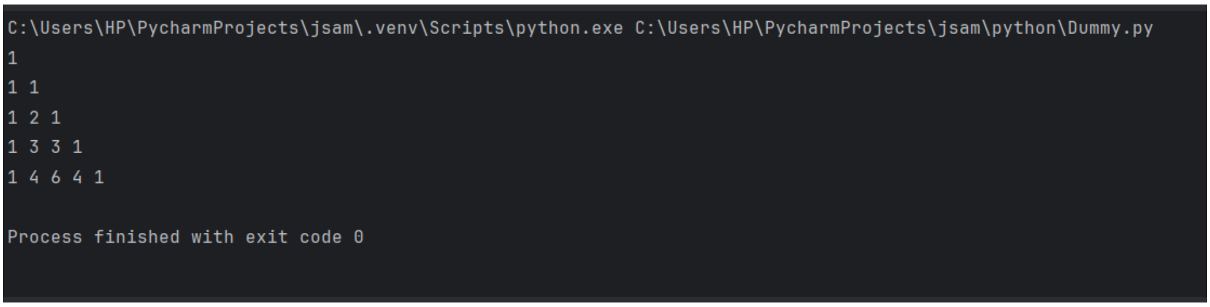
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
1
1   2
1   2   3
1   2   3   4

Process finished with exit code 0
```

4. Program for Pascal's Triangle



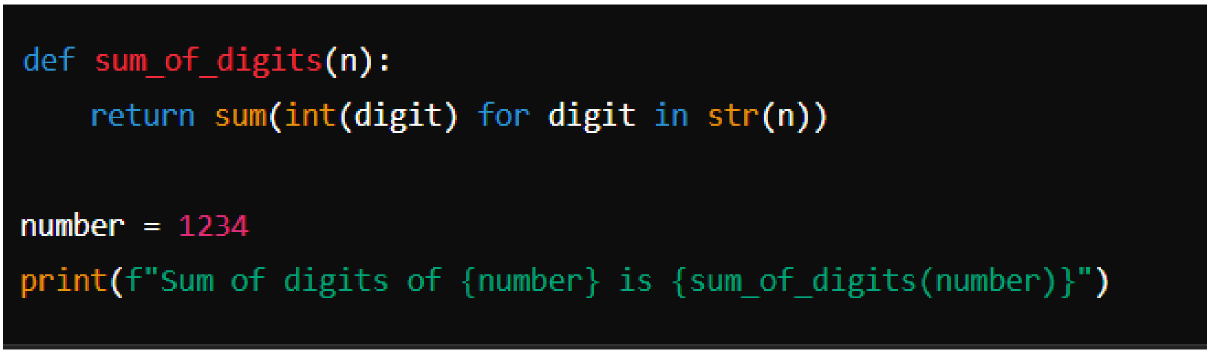
```python
def print_pascals_triangle(n):
    triangle = []
    for i in range(n):
        row = [1] * (i + 1)
        for j in range(1, i):
            row[j] = triangle[i-1][j-1] + triangle[i-1][j]
        triangle.append(row)

    for row in triangle:
        print(" ".join(map(str, row)))

n = 5
print_pascals_triangle(n)
```

OUTPUT:



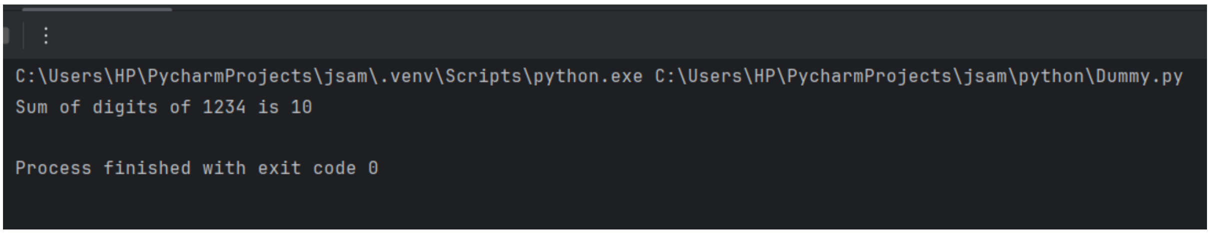
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

Process finished with exit code 0
```

5.Program to Find the Sum of Digits



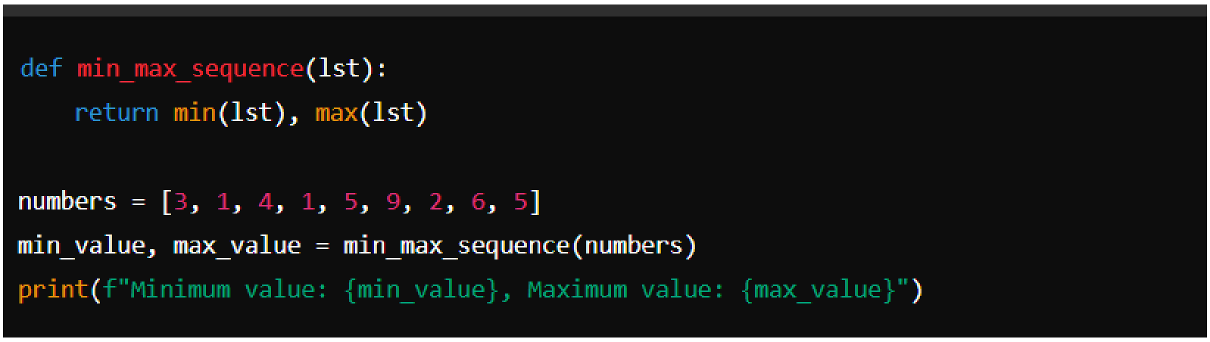
```python
def sum_of_digits(n):
    return sum(int(digit) for digit in str(n))

number = 1234
print(f"Sum of digits of {number} is {sum_of_digits(number)}")
```

OUTPUT:



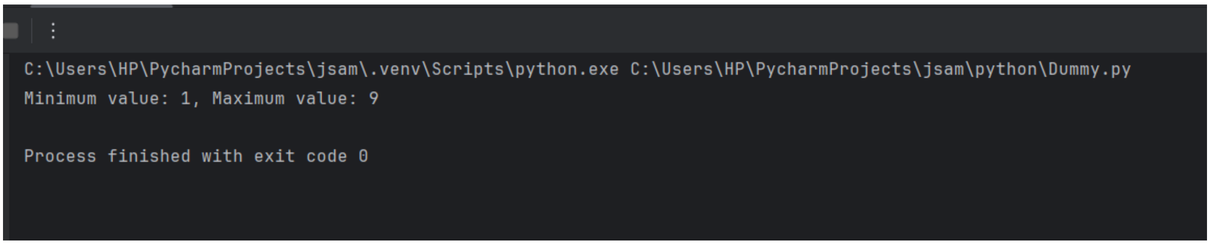
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
Sum of digits of 1234 is 10

Process finished with exit code 0
```

6. Program to Print Minimum and Maximum Value Sequence for All Numbers in a List



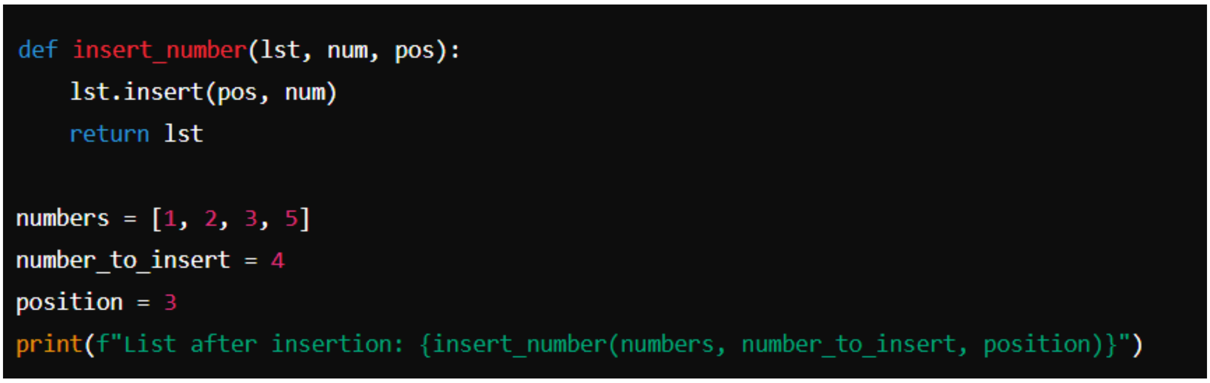
```python
def min_max_sequence(lst):
    return min(lst), max(lst)

numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5]
min_value, max_value = min_max_sequence(numbers)
print(f"Minimum value: {min_value}, Maximum value: {max_value}")
```

OUTPUT:



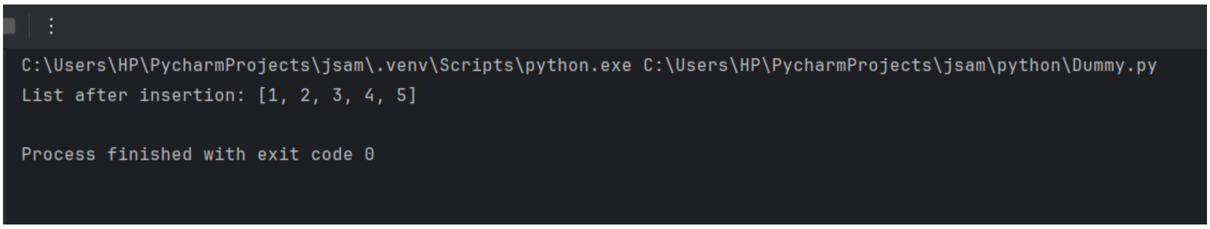
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
Minimum value: 1, Maximum value: 9

Process finished with exit code 0
```

7. Program to Insert a Number in a List



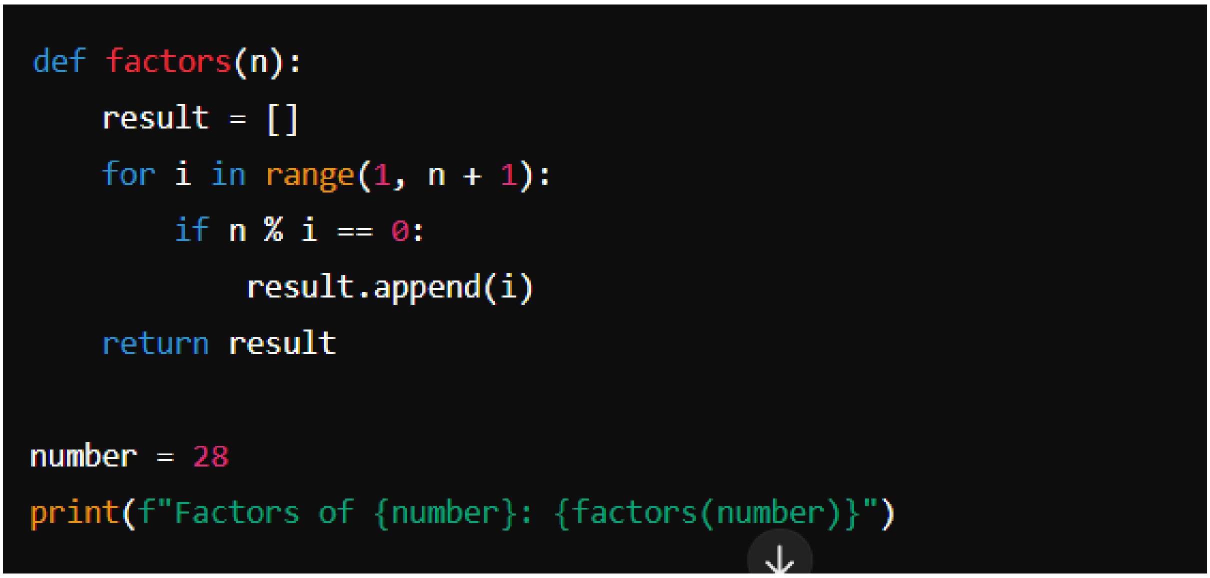
```python
def insert_number(lst, num, pos):
    lst.insert(pos, num)
    return lst

numbers = [1, 2, 3, 5]
number_to_insert = 4
position = 3
print(f"List after insertion: {insert_number(numbers, number_to_insert, position)}")
```

OUTPUT:



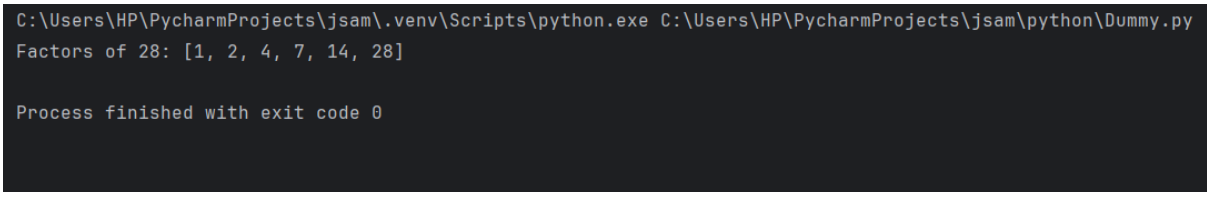
```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
List after insertion: [1, 2, 3, 4, 5]

Process finished with exit code 0
```

8. Program to Generate the List of All Factors for a Given Value



```python
def factors(n):
    result = []
    for i in range(1, n + 1):
        if n % i == 0:
            result.append(i)
    return result

number = 28
print(f"Factors of {number}: {factors(number)}")
```

OUTPUT:



```
C:\Users\HP\PycharmProjects\jsam\.venv\Scripts\python.exe C:\Users\HP\PycharmProjects\jsam\python\Dummy.py
Factors of 28: [1, 2, 4, 7, 14, 28]

Process finished with exit code 0
```