main.py

Share Run

```python
def print_pattern(n):
    for i in range(n, 0, -1):
        for j in range(i):
            print(chr(65 + j), end=' ')
        print()

# Example usage:
n = int(input("Enter the value of n: "))
print_pattern(n)
```

Output Clear

```
Enter the value of n: 5
A B C D E
A B C D
A B C
A B
A

=== Code Execution Successful ===
```

main.py

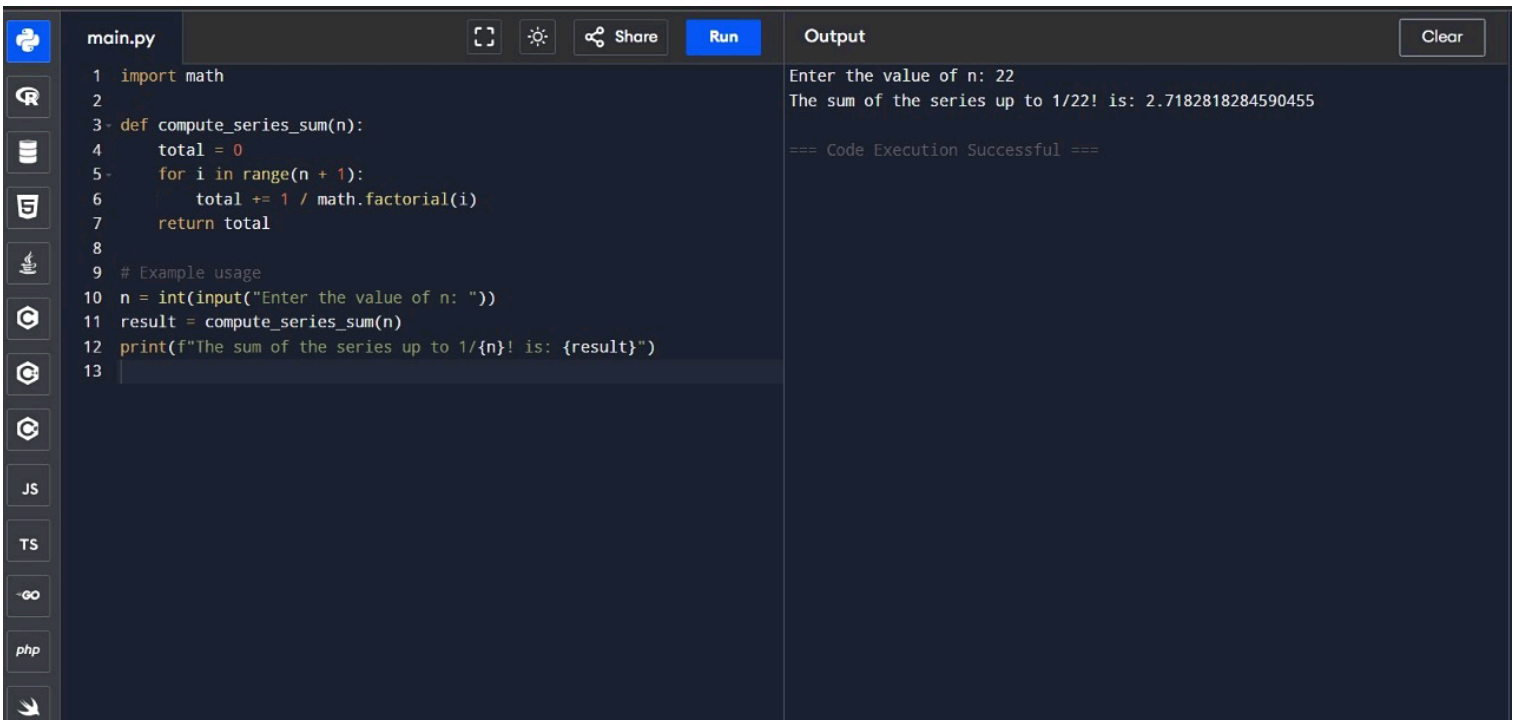
Share Run

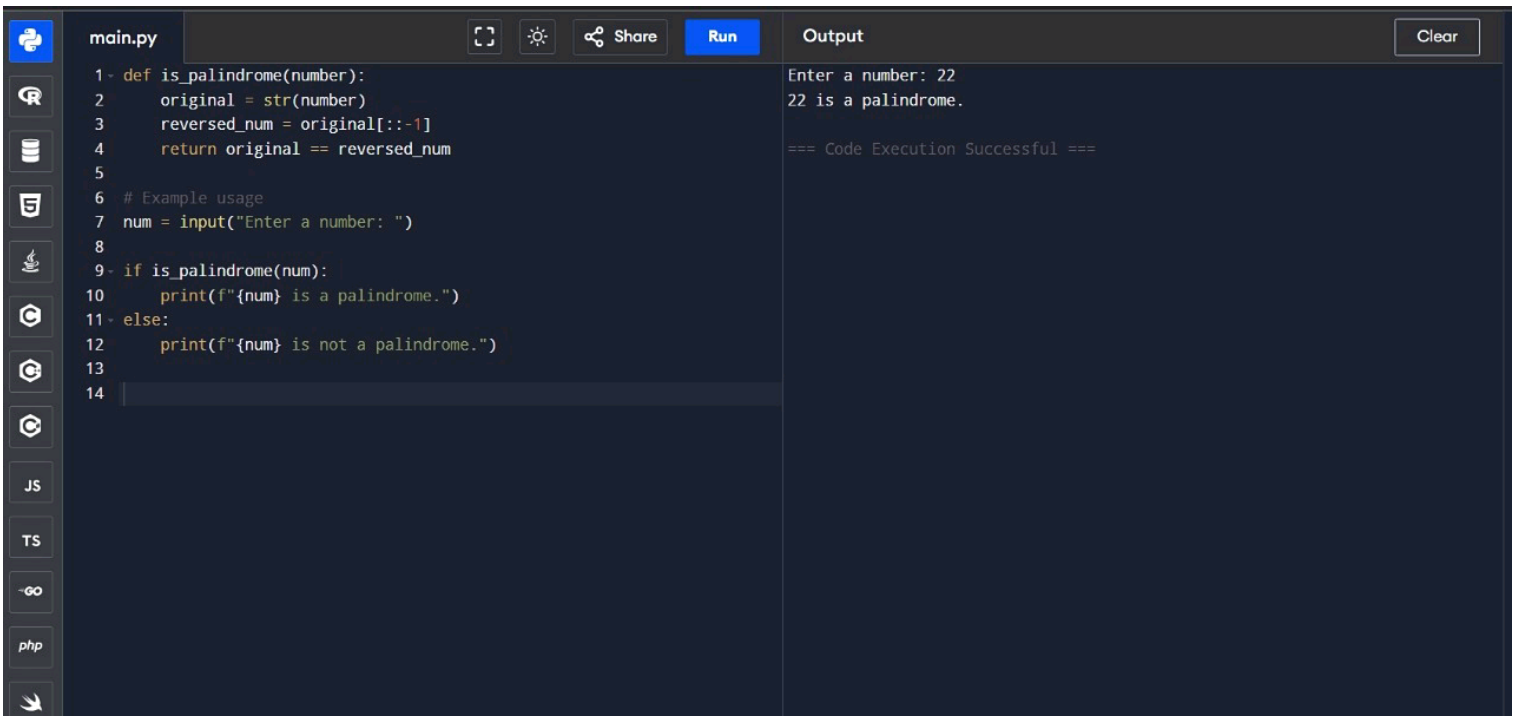
```python
import math

def cosine_series(x, n):
    cos_x = 0
    for i in range(n):
        term = ((-1) ** i) * (x ** (2 * i)) / math.factorial(2 * i)
        cos_x += term
    return cos_x

# Example usage
x = float(input("Enter the value of x (in radians): "))
n = int(input("Enter the number of terms in the series: "))

result = cosine_series(x, n)
print(f"Cosine approximation of cos({x}) using {n} terms is: {result}"
    )
print(f"Actual value using math.cos: {math.cos(x)}")

```

Output

Clear

```
Enter the value of x (in radians): 6
Enter the number of terms in the series: 5
Cosine approximation of cos(6.0) using 5 terms is: 13.857142857142861
Actual value using math.cos: 0.960170286650366

=== Code Execution Successful ===
```


main.py
Share
Run
Output
Clear
import math
def compute_series_sum(n):
total = 0
for i in range(n + 1):
total += 1 / math.factorial(i)
return total
# Example usage
n = int(input("Enter the value of n: "))
result = compute_series_sum(n)
print(f"The sum of the series up to 1/{n}! is: {result}")
Enter the value of n: 22
The sum of the series up to 1/22! is: 2.7182818284590455
=== Code Execution Successful ===


main.py
Share
Run
Output
Clear
def is_palindrome(number):
original = str(number)
reversed_num = original[::-1]
return original == reversed_num
# Example usage
num = input("Enter a number: ")
if is_palindrome(num):
print(f"{num} is a palindrome.")
else:
print(f"{num} is not a palindrome.")
Enter a number: 22
22 is a palindrome.
=== Code Execution Successful ===

main.py

Share Run

```python
def calculate_simple_interest(principal, years, is_senior):
    if principal <= 0:
        return "Principal must be greater than 0."
    if years <= 0:
        return "Number of years must be greater than 0."

    rate = 0.12 if is_senior else 0.10
    interest = principal * rate * years
    return f"Interest: {interest}"

# Get user input
try:
    principal = float(input("Enter the principal amount: "))
    years = int(input("Enter the no of years: "))
    senior = input("Is customer senior citizen (y/n): ").strip
        ().lower()

    if senior not in ['y', 'n']:
        print("Invalid input for senior citizen status.")
    else:
        is_senior = senior == 'y'
        result = calculate_simple_interest(principal, years,
            is_senior)
        print(result)

except ValueError:
```

Output Clear

```
Enter the principal amount: 200000
Enter the no of years: 2
Is customer senior citizen (y/n): n
Interest: 40000.0

=== Code Execution Successful ===
```

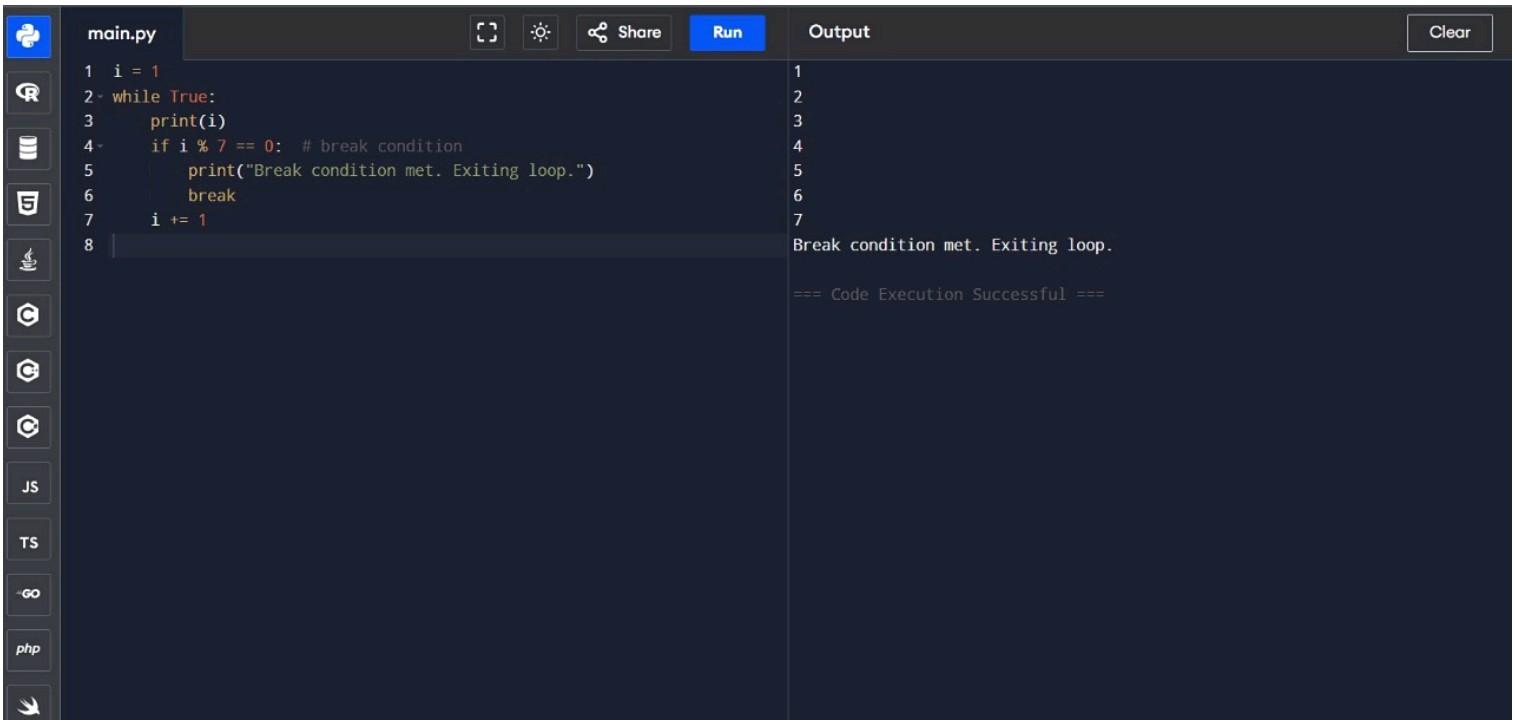
main.py

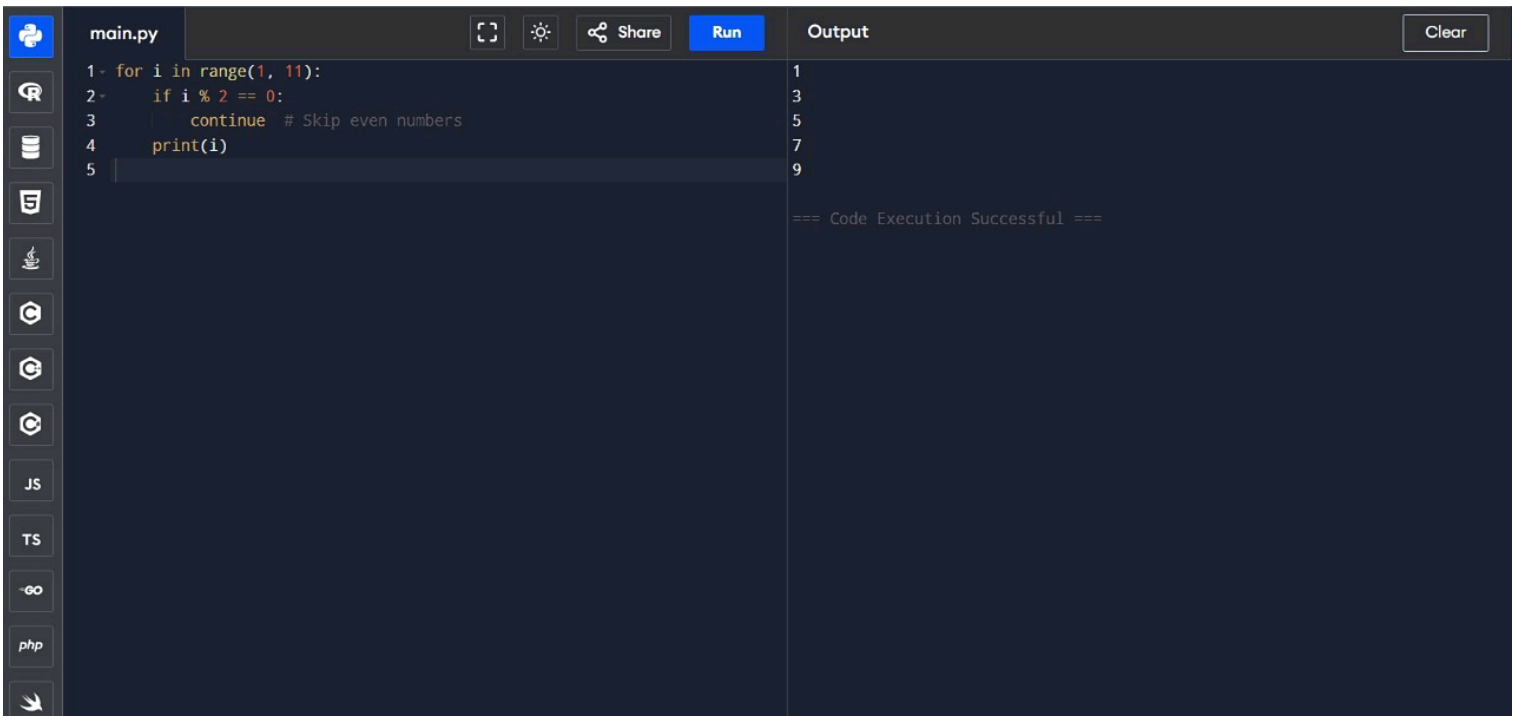
Share Run

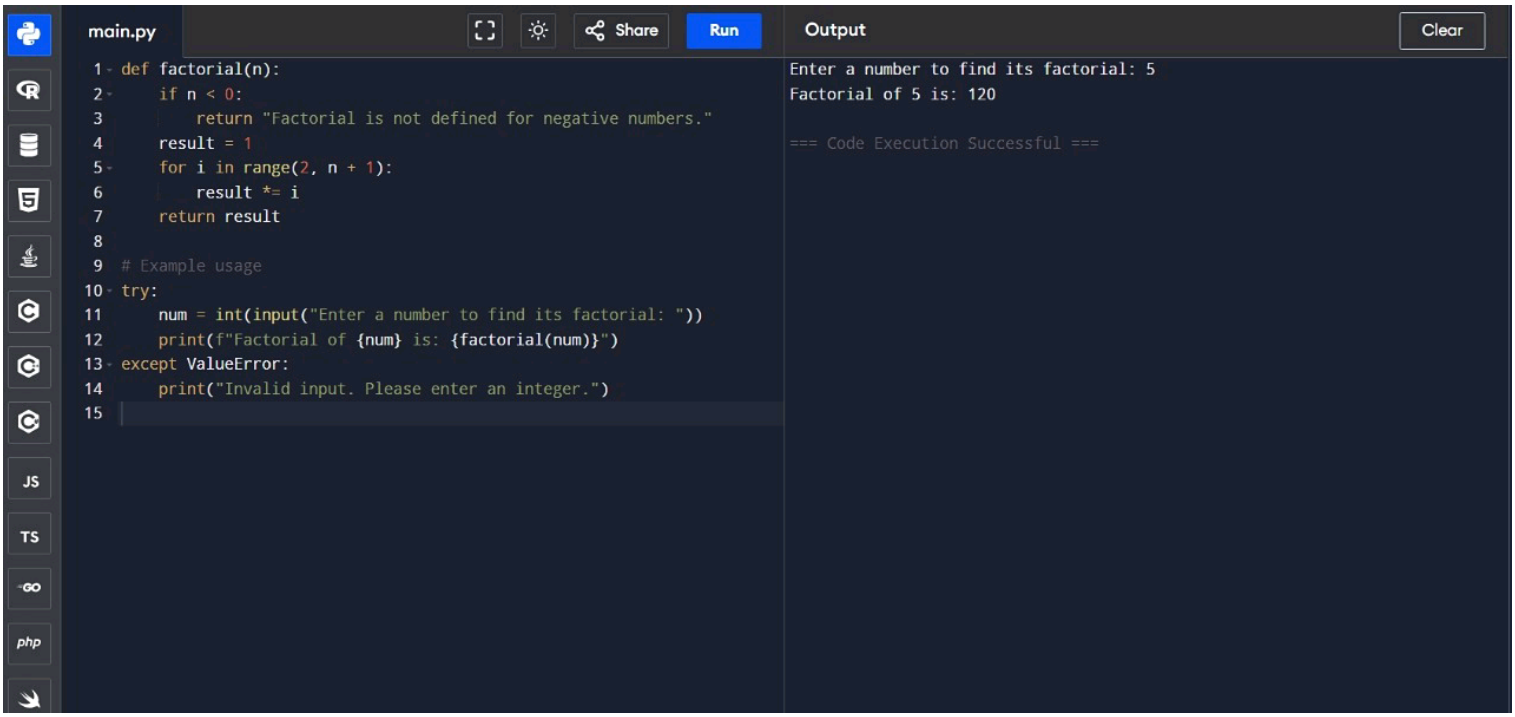
```python
def sumsquare(l):
    odd = sum(x**2 for x in l if x % 2 != 0)
    even = sum(x**2 for x in l if x % 2 == 0)
    return [odd, even]

# Input handling
try:
    n = int(input("Enter the number of elements: "))
    if n <= 0:
        print("Please enter a positive number of elements.")
    else:
        numbers = []
        for _ in range(n):
            num = int(input("Enter the element: "))
            numbers.append(num)
        result = sumsquare(numbers)
        print("Output:")
        print(result)
except ValueError:
    print("Invalid input. Please enter integers only.")

```

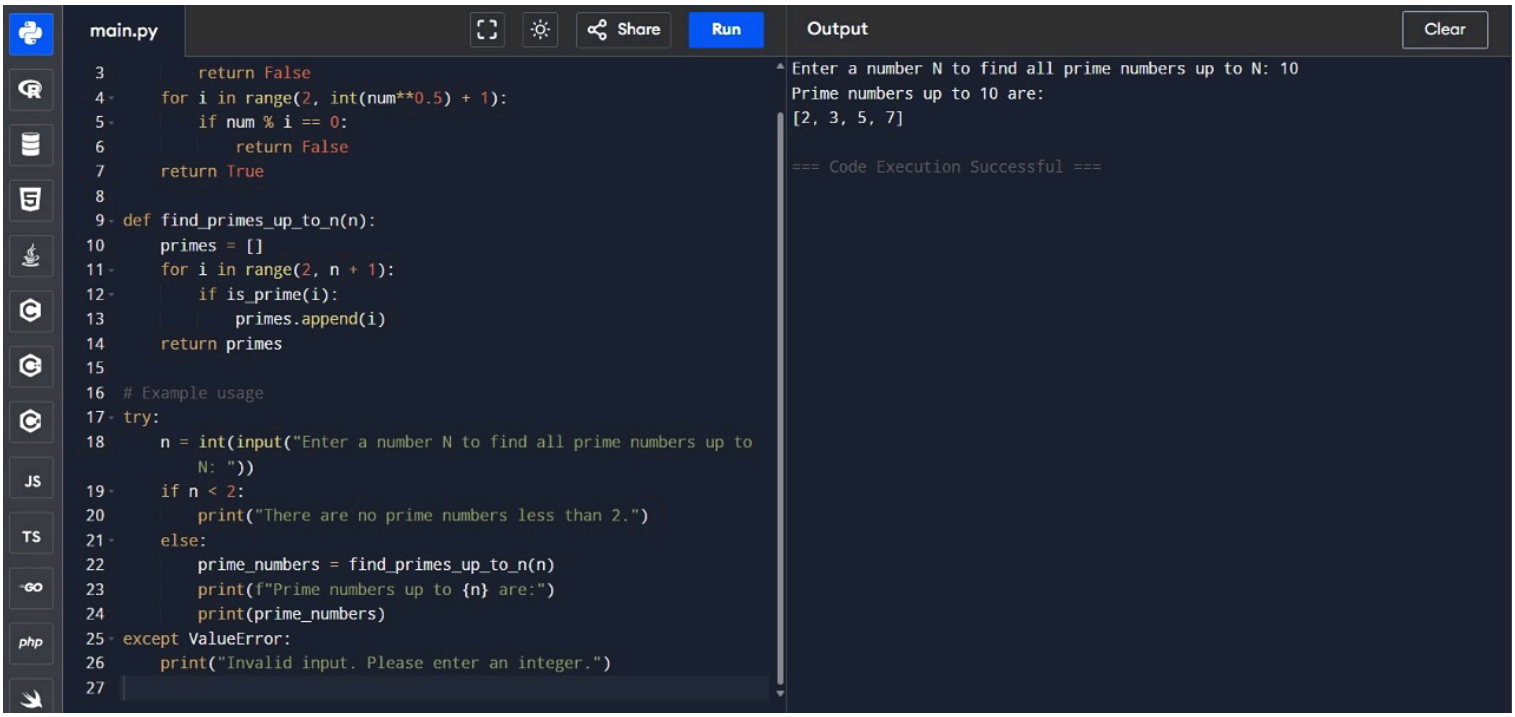
Output Clear

```
Enter the number of elements: 7
Enter the element: 18
Enter the element: 9
Enter the element: 1
Enter the element: 12
Enter the element: 13
Enter the element: 4
Enter the element: 30
Output:
[251, 1384]

=== Code Execution Successful ===
```


main.py
Share
Run
Output
Clear
1 i = 1
2 while True:
3 print(i)
4 if i % 7 == 0: # break condition
5 print("Break condition met. Exiting loop.")
6 break
7 i += 1
8
1
2
3
4
5
6
7
Break condition met. Exiting loop.
=== Code Execution Successful ===
JS
TS
GO
php


main.py
Share
Run
Output
Clear
for i in range(1, 11):
if i % 2 == 0:
continue # Skip even numbers
print(i)
1
3
5
7
9
=== Code Execution Successful ===


main.py
Share
Run
Output
Clear
def factorial(n):
if n < 0:
return "Factorial is not defined for negative numbers."
result = 1
for i in range(2, n + 1):
result *= i
return result
# Example usage
try:
num = int(input("Enter a number to find its factorial: "))
print(f"Factorial of {num} is: {factorial(num)}")
except ValueError:
print("Invalid input. Please enter an integer.")
Enter a number to find its factorial: 5
Factorial of 5 is: 120
=== Code Execution Successful ===


main.py
Share
Run
Output
Clear
return False
for i in range(2, int(num**0.5) + 1):
if num % i == 0:
return False
return True
def find_primes_up_to_n(n):
primes = []
for i in range(2, n + 1):
if is_prime(i):
primes.append(i)
return primes
# Example usage
try:
n = int(input("Enter a number N to find all prime numbers up to N: "))
if n < 2:
print("There are no prime numbers less than 2.")
else:
prime_numbers = find_primes_up_to_n(n)
print(f"Prime numbers up to {n} are:")
print(prime_numbers)
except ValueError:
print("Invalid input. Please enter an integer.")
Enter a number N to find all prime numbers up to N: 10
Prime numbers up to 10 are:
[2, 3, 5, 7]
=== Code Execution Successful ===

main.py

Share Run

```python
def print_pattern(n):
    for i in range(1, n + 1):  # Outer loop to control rows
        for j in range(i):  # Inner loop to control the number of stars in each row
            print("*", end=" ")
        print()  # Move to the next line after each row

# Example usage
n = int(input("Enter the number of rows: "))
print_pattern(n)
```

Output

Clear

```
Enter the number of rows: 7
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *

=== Code Execution Successful ===
```