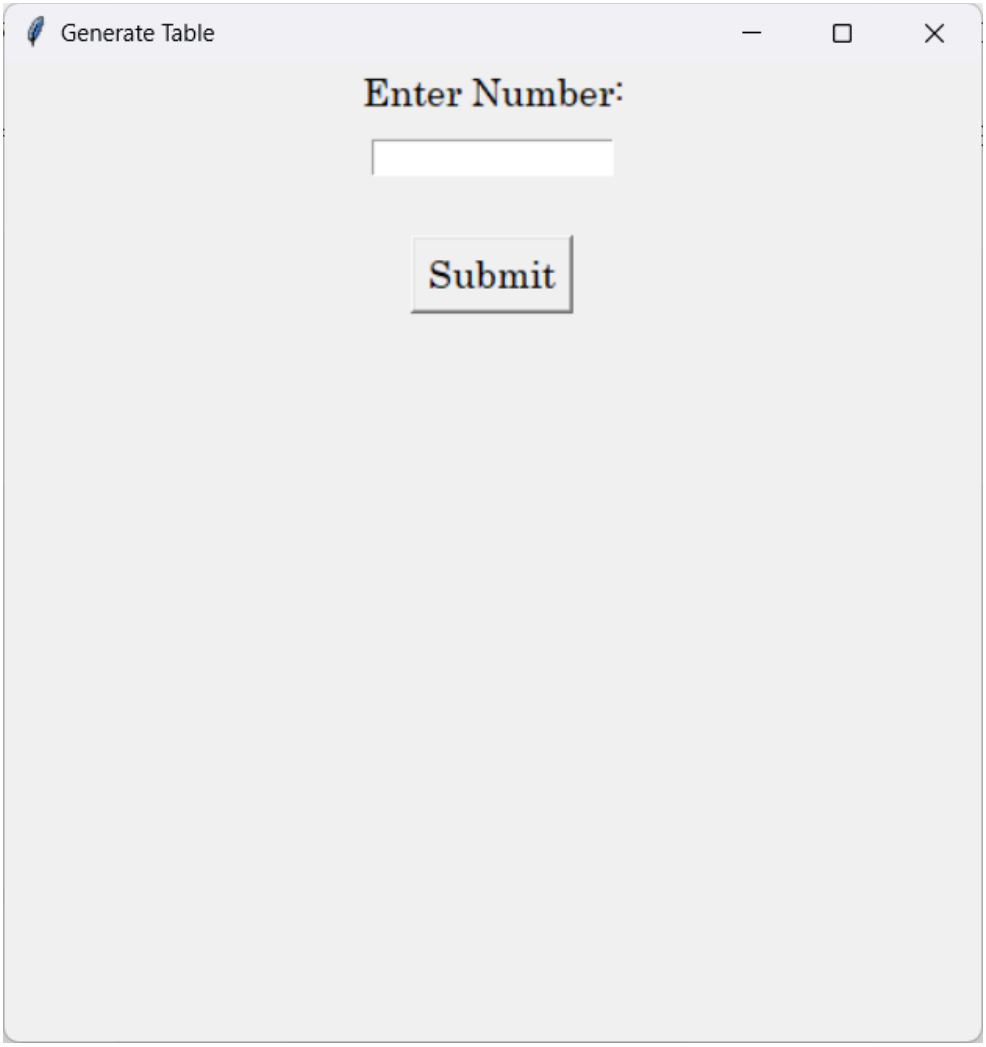
Name: Mohmadhusen Ahmadbhai Khimani
Enrollment No: 22FOTCA11071
Roll No: 14
Div: 6-BCA-B
Subject: Python Programming
Subject Code: BCA619

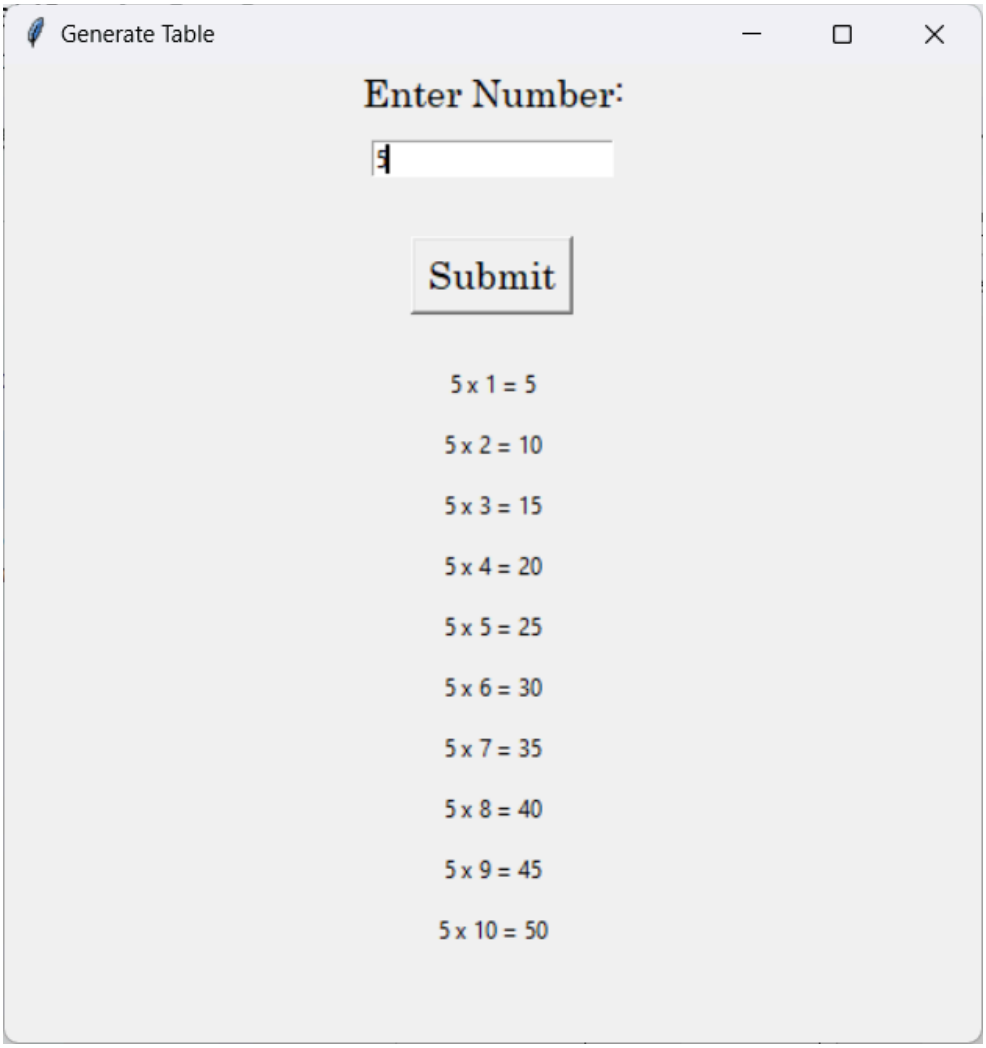
Tutorial 13

Q.1	**Table**
Code:	#Name: Mohmadhusen Khimani #Enrollment No: 22FOTCA11071 #Datatypes for GUI IntVar() DoubleVar() BooleanVar() StringVar() #Datatypes in Console int,float,string,commplex,boolean #Q.1 Genrate the table of entered number by the user. from tkinter import * f = Tk() f.geometry("500x500") f.title("Generate Table") Label(f, text="Enter Number:",

```
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

def submit():
    num = int(val.get())
    for i in range(1,11):
       Label(f,text=f"{num} x {i} = {num *
i}").pack(pady=5)

Button(f, text="Submit", font=("Century", 14),
command=submit).pack(pady=20)

f.mainloop()
```

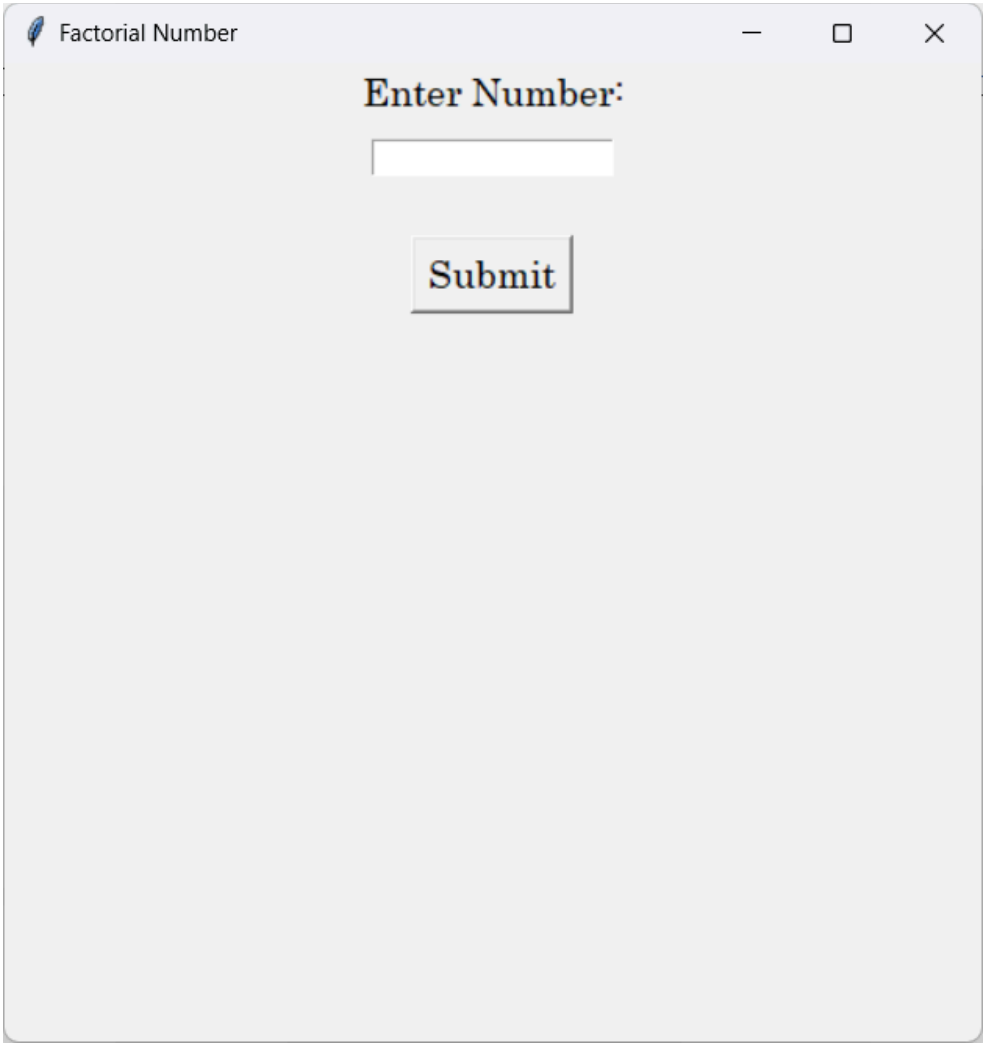
Output:

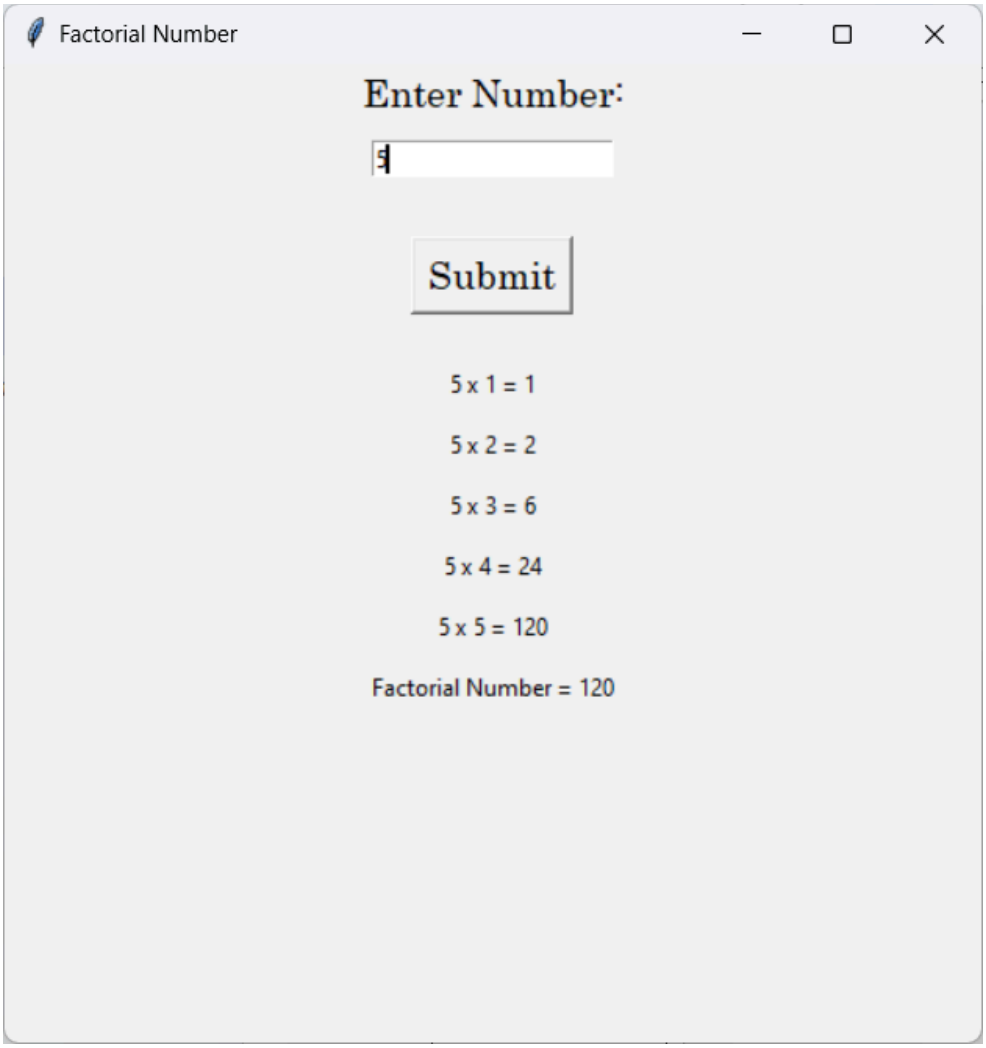




Q.2	**Factorial Number**
Code:	#Name: Mohmadhusen Khimani #Enrollment No: 22FOTCA11071 #Q.2 Factorial Number from tkinter import * f = Tk()

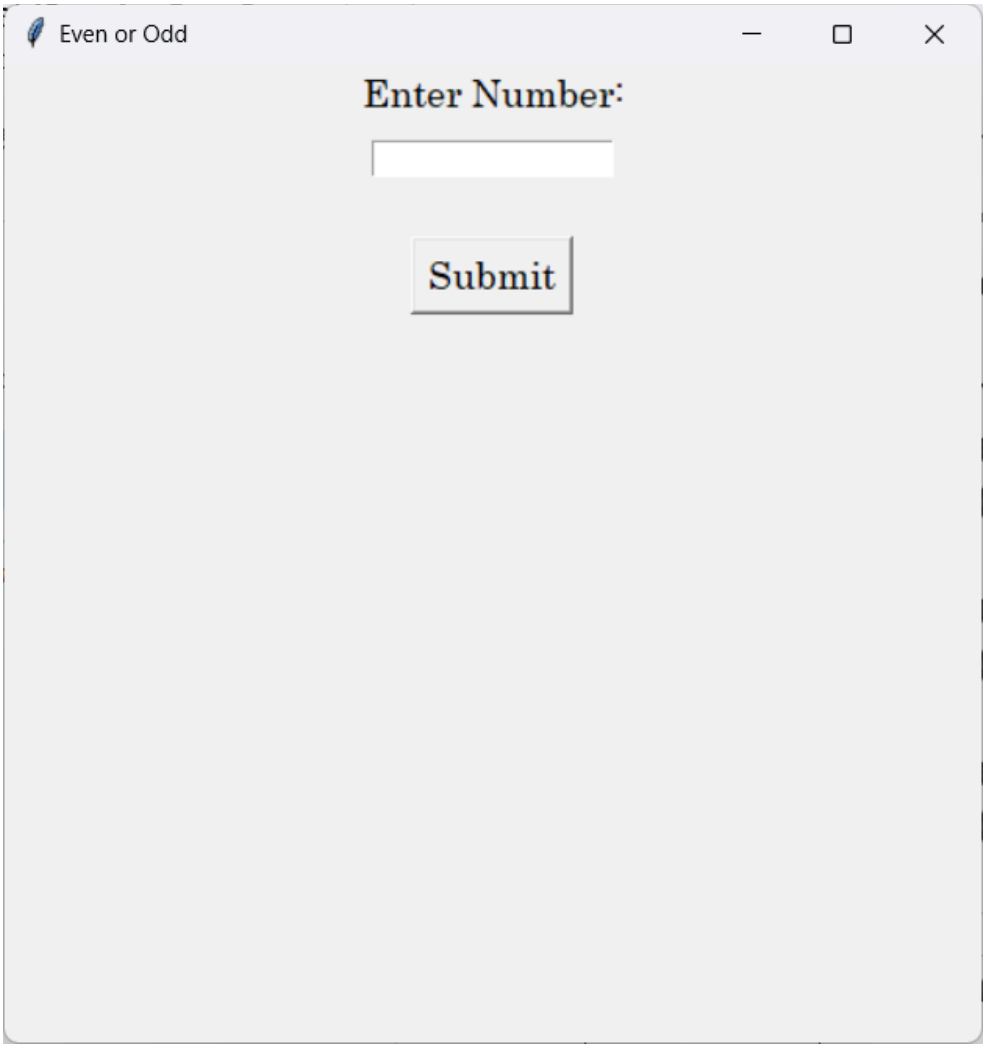
```
f.geometry("500x500")
f.title("Factorial Number")

Label(f, text="Enter Number:",
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

def submit():
    num = int(val.get())
    fact=1
    for i in range(1,(num+1),1):
        fact= fact * i
        Label(f,text=f"{num} x {i} =
{fact}").pack(pady=5)

    Label(f,text=f"Factorial Number =
{fact}").pack(pady=5)

Button(f, text="Submit", font=("Century", 14),
command=submit).pack(pady=20)

f.mainloop()
```

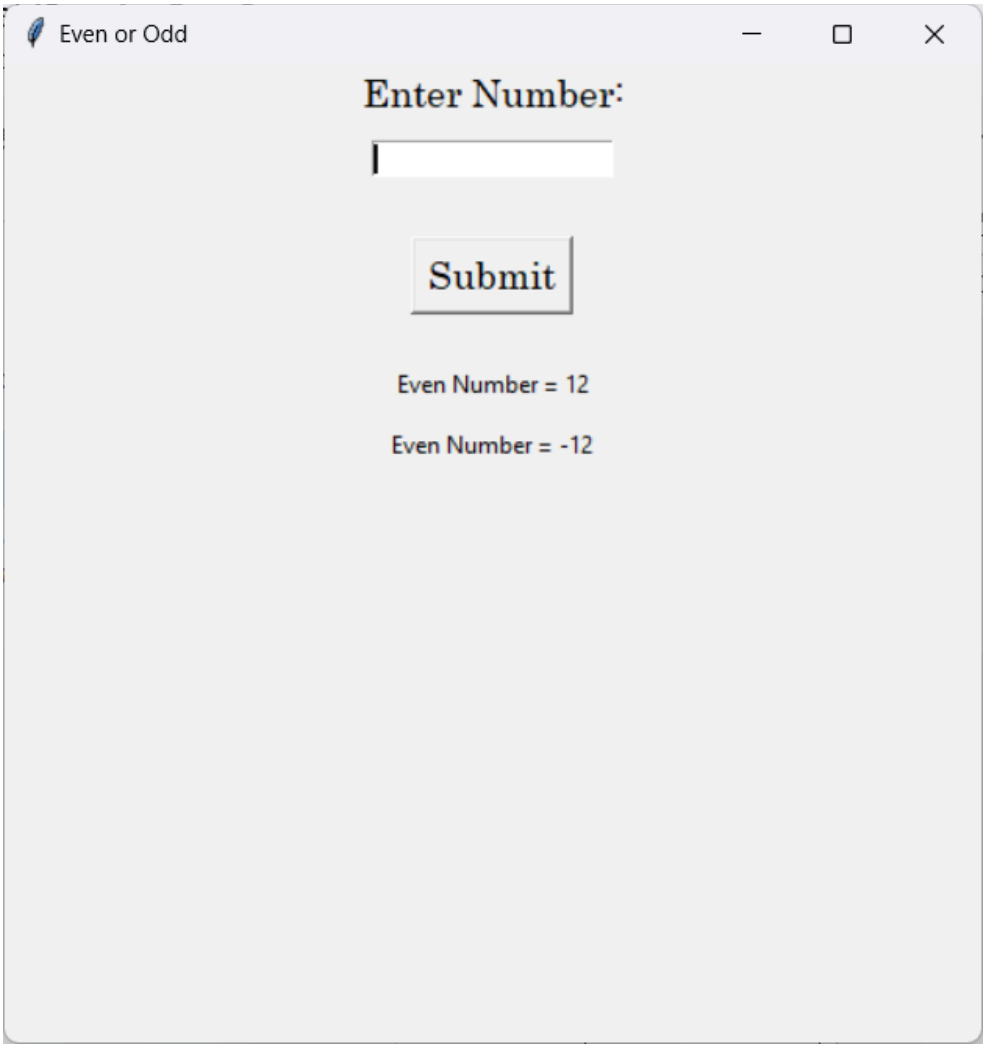
Output:	



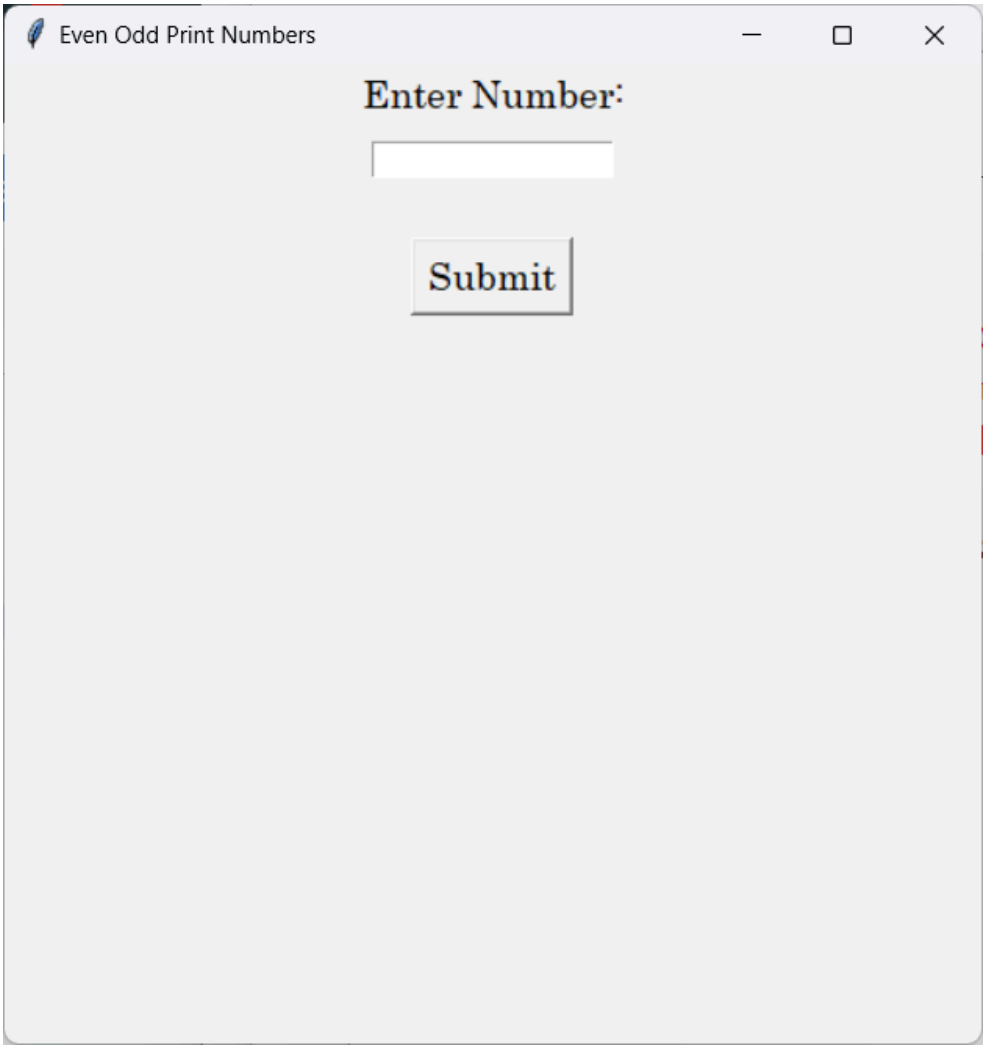
Q.3	**Even or Odd**
Code:	#Name: Mohmadhusen Khimani #Enrollment No: 22FOTCA11071 #Q.3 Even or Odd from tkinter import * f = Tk()

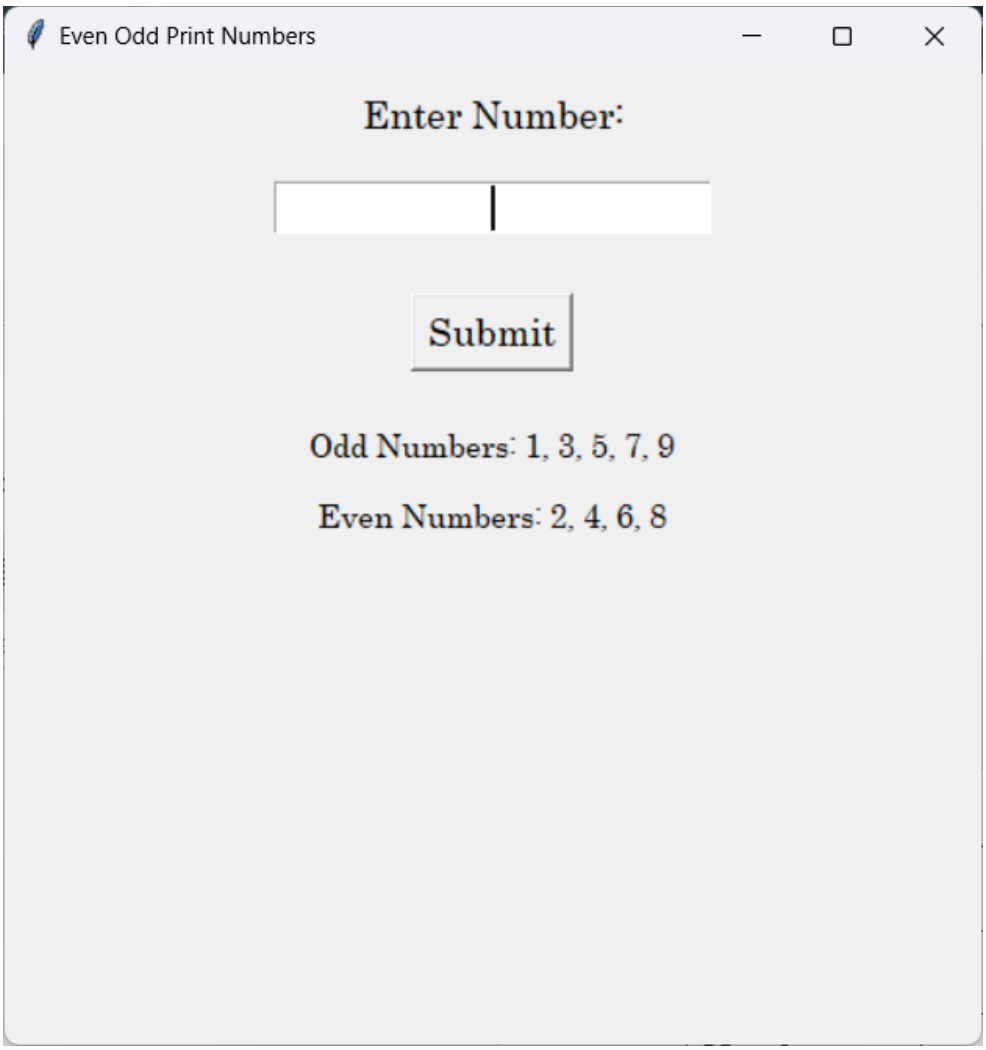
```
f.geometry("500x500")
f.title("Even or Odd")

Label(f, text="Enter Number:",
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

def submit():
    num = int(val.get())
    if num == 0:
        Label(f,text=f"Zero =
{num}").pack(pady=5)
    elif num % 2 == 0:
        Label(f,text=f"Even Number =
{num}").pack(pady=5)
    else:
        Label(f,text=f"Odd Number =
{num}").pack(pady=5)

Button(f, text="Submit", font=("Century", 14),
command=submit).pack(pady=20)

f.mainloop()
```

Output:	

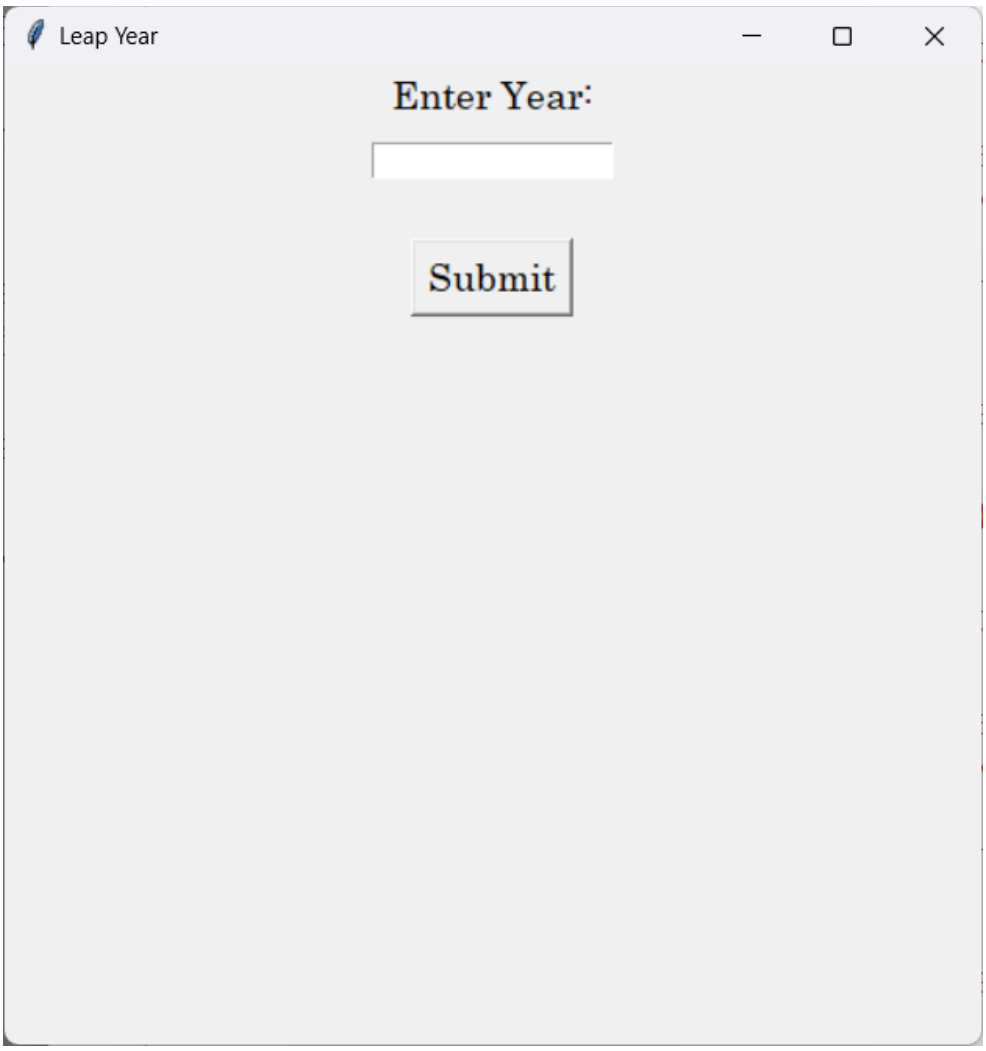
	
Q.4	**Even Odd .**
Code:	#Name: Mohmadhusen Khimani #Enrollment No: 22FOTCA11071 #Q.4 You need to print digit. If I enter 6 so, 2,4,6,8,10,12 Even and If enter 5 so, 1,3,5,7,9 Odd. from tkinter import *

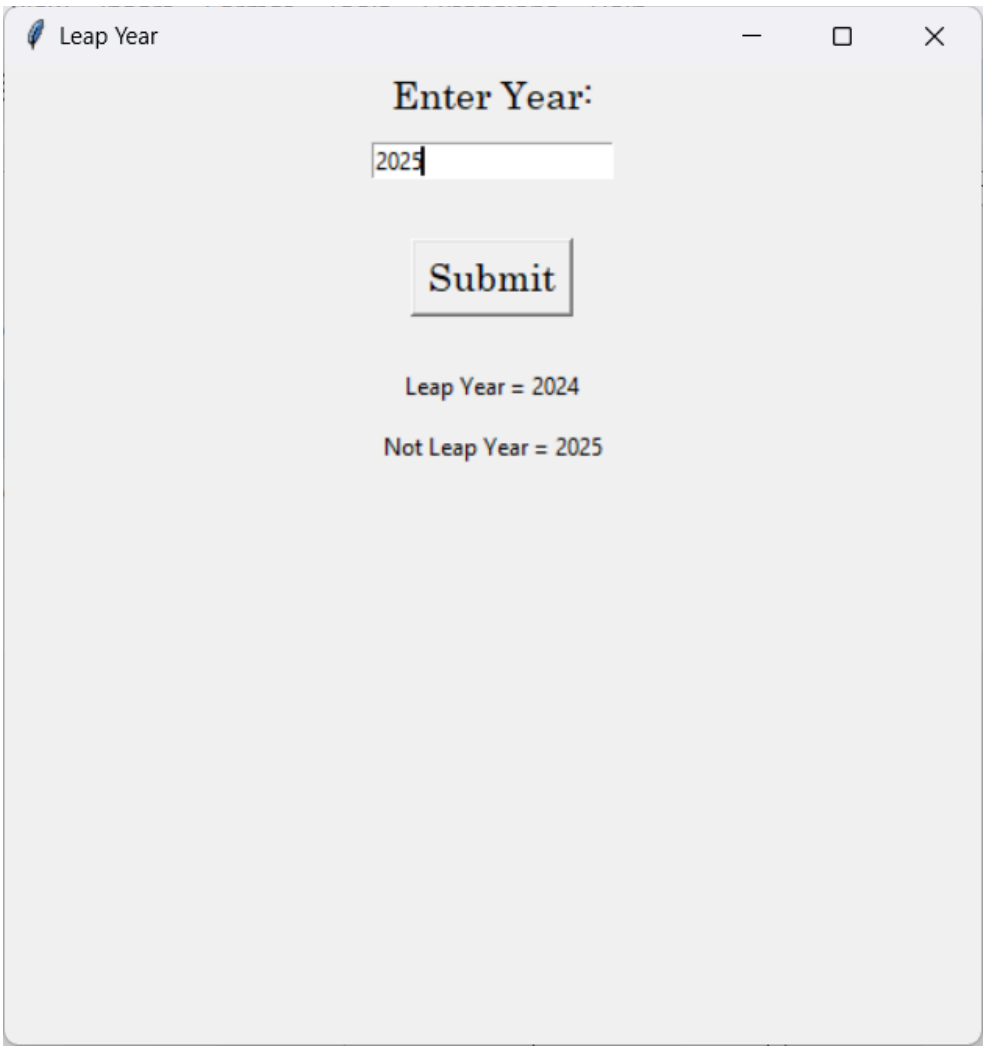
```
f = Tk()
f.geometry("500x500")
f.title("Even Odd Print Numbers")

Label(f, text="Enter Number:",
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

def submit():
    num = int(val.get())
    even=0
    odd=0
    for i in range(1(num+1),1):
        if num % i == 0:
            even+=1;
        else:
            odd+=1;


    Label(f,text=f"Even Number =
{even}").pack(pady=5)
    Label(f,text=f"Odd Number =
{odd}").pack(pady=5)


Button(f, text="Submit", font=("Century", 14),
command=submit).pack(pady=20)
```

	f.mainloop()
Output:	

	
Q.5	**Leap Year**
Code:	#Name: Mohmadhusen Khimani #Enrollment No: 22FOTCA11071 #Q.5 Leap Year from tkinter import *

```
f = Tk()
f.geometry("500x500")
f.title("Leap Year")

Label(f, text="Enter Year:", font=("Century",
14)).pack()
val = Entry(f)
val.pack(pady=10)

def submit():
    year = int(val.get())

    if year <= 0:
        Label(f,text=f"Invalid =
{year}").pack(pady=5)
    elif year % 4 == 0:
        Label(f,text=f"Leap Year =
{year}").pack(pady=5)
    else:
        Label(f,text=f"Not Leap Year =
{year}").pack(pady=5)

Button(f, text="Submit", font=("Century", 14),
command=submit).pack(pady=20)

f.mainloop()
```

Output:	

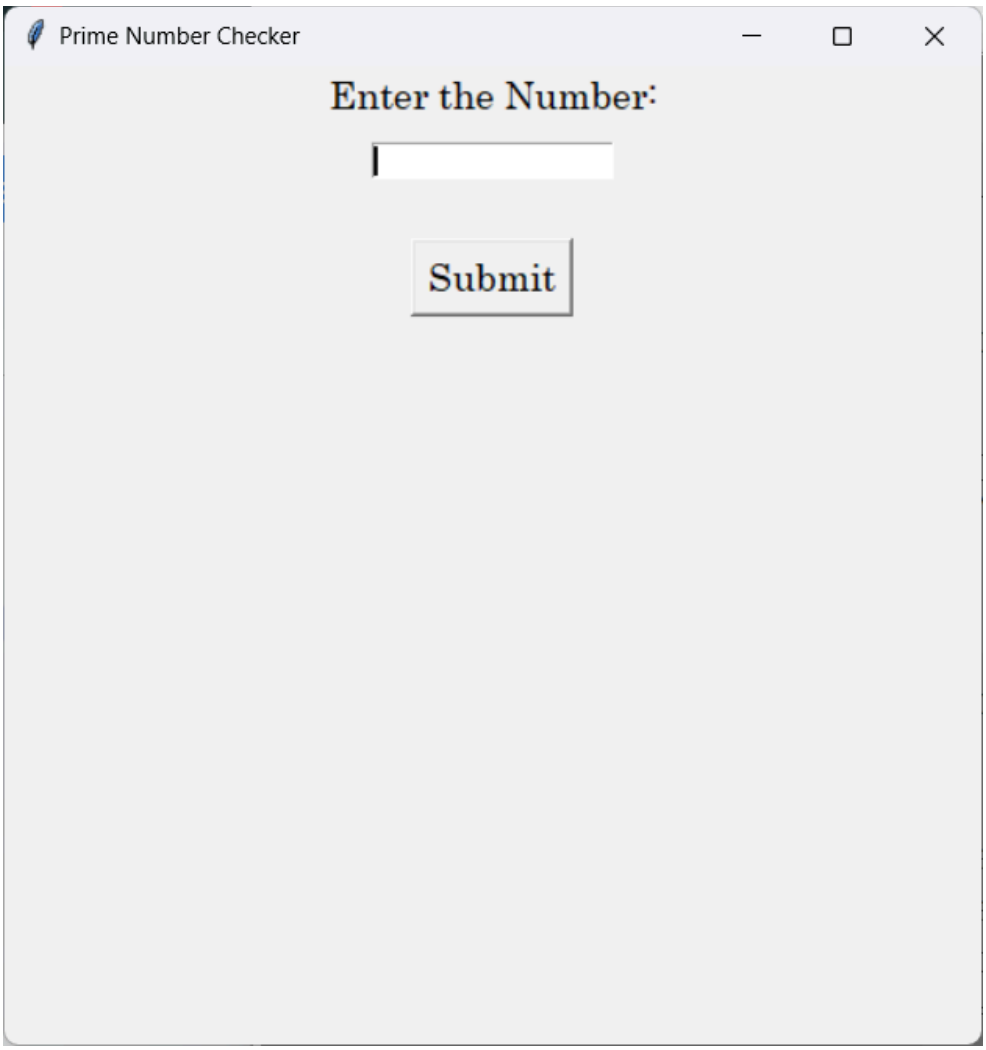


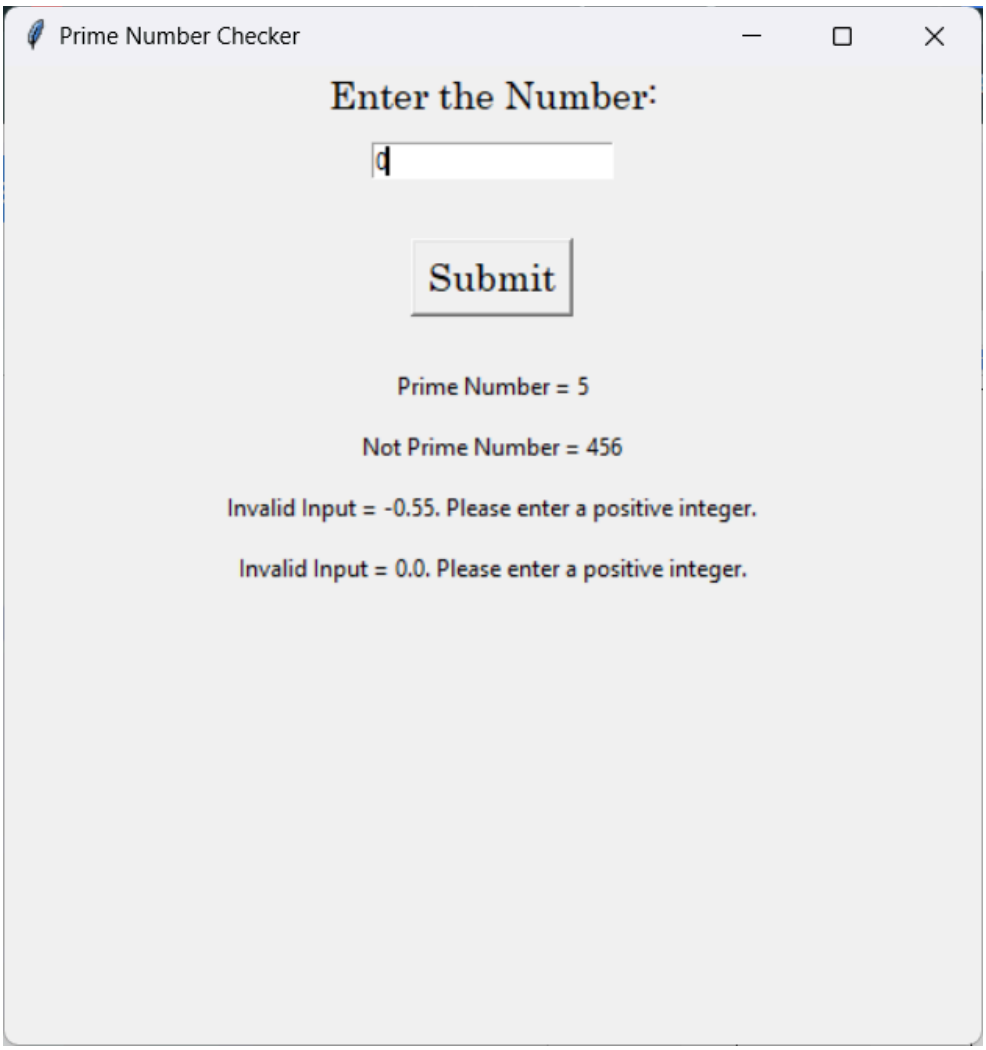
Q.6	**Prime Number**

Code:	from tkinter import *
	
	# Q.6 Create the main application window
	f = Tk()
	f.geometry("500x500")
	f.title("Prime Number Checker")

```
# Add a label and entry field for user input
Label(f, text="Enter the Number:",
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

# Function to handle the button click
def submit():
    try:
        # Get and validate the input
        num = float(val.get())
        if num <= 0 or not num.is_integer():
            Label(f, text=f"Invalid Input = {num}.
Please enter a positive
integer.").pack(pady=5)
            return

        # Convert to integer
        num = int(num)

        # Prime number logic
        if num < 2:
            Label(f, text=f"Not Prime Number =
{num}").pack(pady=5)
        else:
            is_prime = True
            for i in range(2, int(num ** 0.5) + 1):
                if num % i == 0:
                    is_prime = False
                    break
```

```
            if is_prime:
                Label(f, text=f"Prime Number = 
{num}").pack(pady=5)
            else:
                Label(f, text=f"Not Prime Number = 
{num}").pack(pady=5)
    except ValueError:
        # Handle non-numeric inputs
        Label(f, text="Invalid Input. Please enter 
a valid number.").pack(pady=5)

# Add a submit button
Button(f, text="Submit", font=("Century", 14), 
command=submit).pack(pady=20)

# Start the application loop
f.mainloop()
```

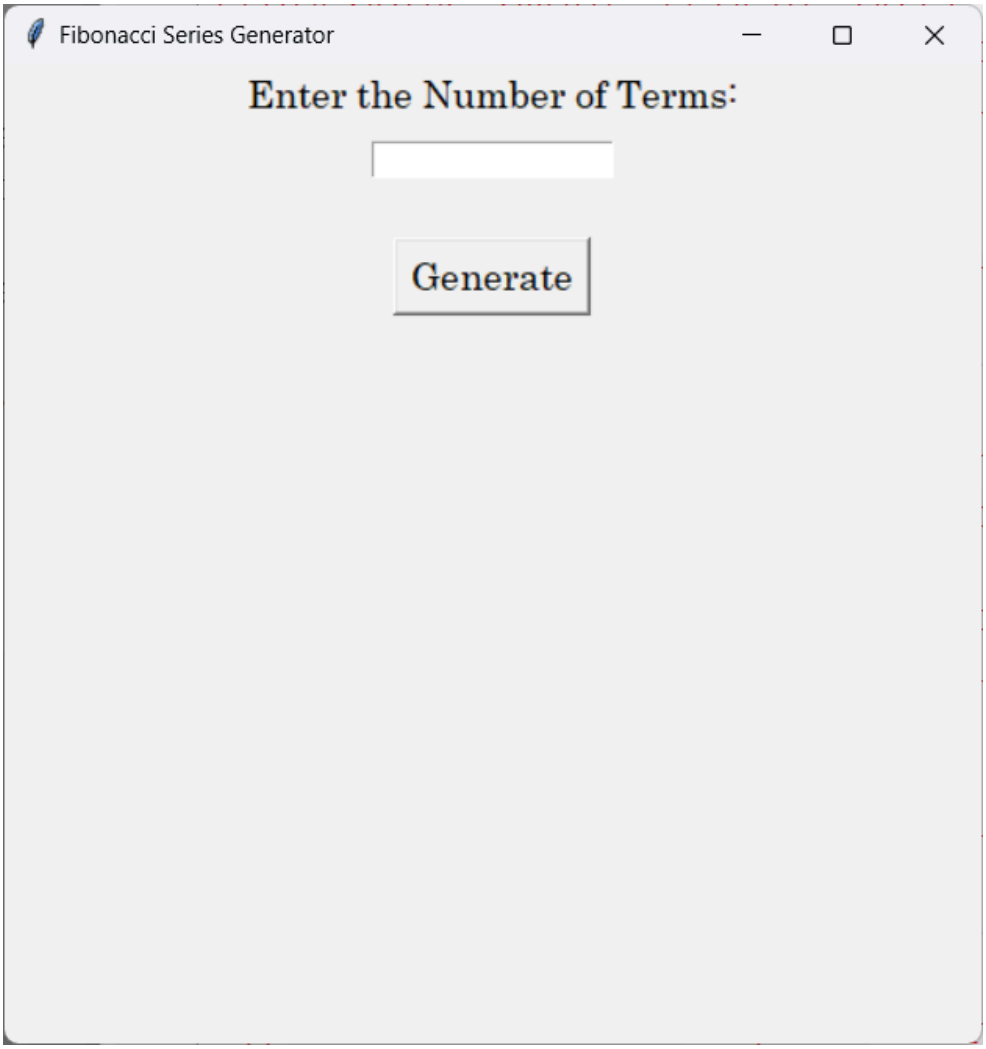
Output:	

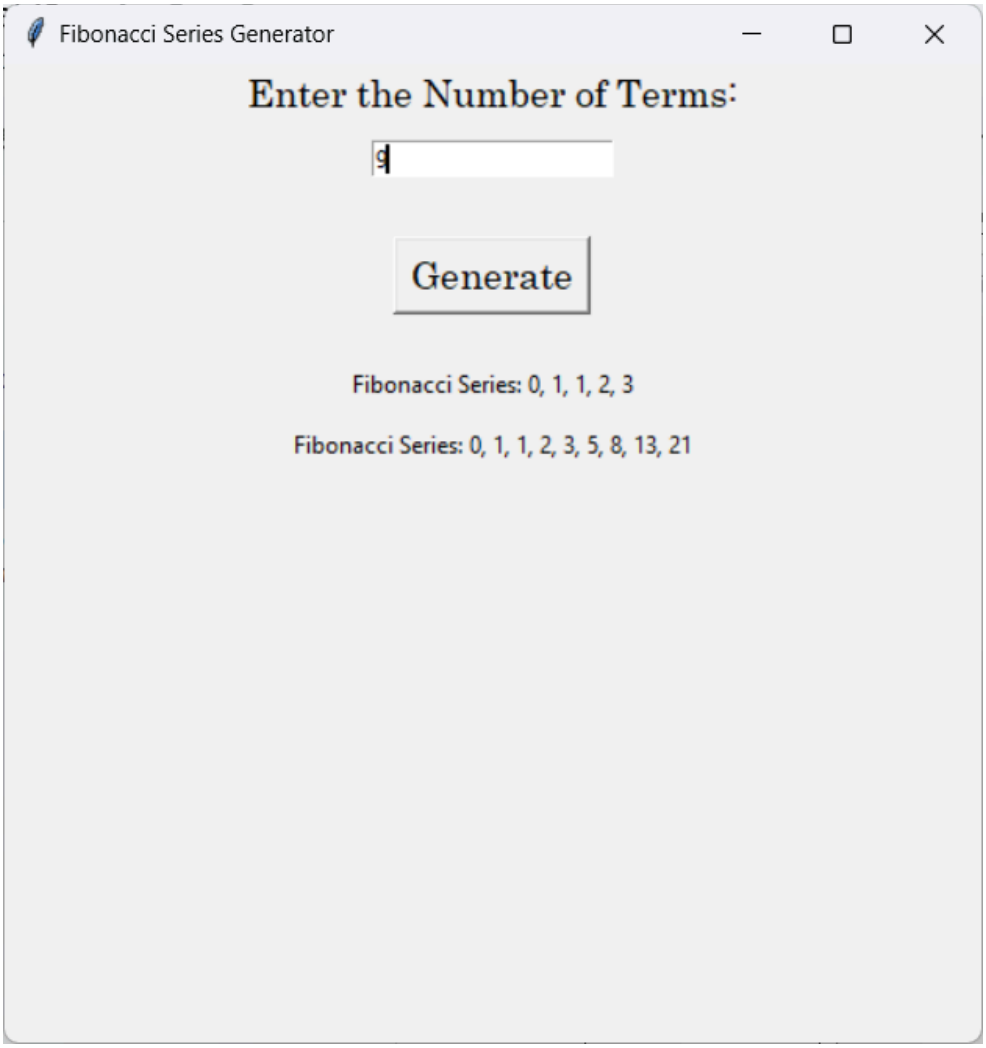


Q.7	**Fibonacci Series**
Code:	#Q.7 Fibonacci Series from tkinter import * f = Tk() f.geometry("500x500") f.title("Fibonacci Series Generator")

```
Label(f, text="Enter the Number of Terms:",
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

def generate_fibonacci():
    try:
        terms = int(val.get())
        if terms <= 0:
            Label(f, text=f"Invalid Input = {terms}.
Please enter a positive
integer.").pack(pady=5)
            return

        fibonacci_series = []
        a, b = 0, 1
        for _ in range(terms):
            fibonacci_series.append(a)
            a, b = b, a + b

        Label(f, text=f"Fibonacci Series: {',
'.join(map(str,
fibonacci_series))}").pack(pady=5)
    except ValueError:
        Label(f, text="Invalid Input. Please enter
a valid number.").pack(pady=5)

Button(f, text="Generate", font=("Century",
14),
command=generate_fibonacci).pack(pady=20
```

	)
	
	# Start the application loop
	f.mainloop()

Output:	



Q.8	**Patterns**

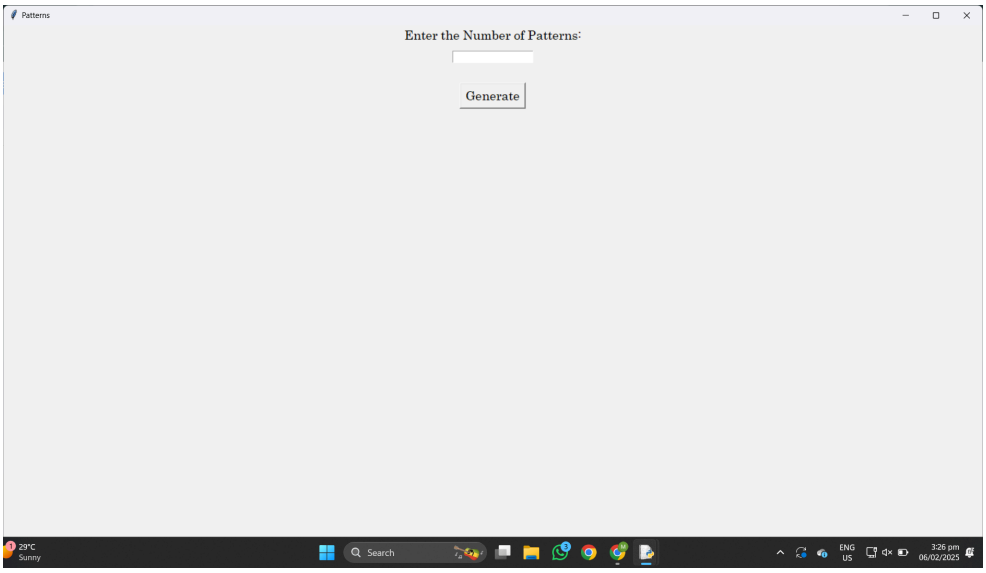
Code:	from tkinter import *
	
	# Initialize the main application window
	f = Tk()
	f.geometry("1500x1500")
	f.title("Patterns")
	
	# Label and input box for entering the number of patterns

```
Label(f, text="Enter the Number of Patterns:",
font=("Century", 14)).pack()
val = Entry(f)
val.pack(pady=10)

def generate():
    # Clear all previous patterns before
generating new ones
    for widget in f.winfo_children():
        if isinstance(widget, Label) and widget !=
val:
            widget.destroy()

    try:
        # Get the number of patterns to display
        num_patterns = int(val.get())
        if num_patterns < 1 or num_patterns > 5:
            raise ValueError("Enter a number
between 1 and 5.")

        patterns = []

        # Define the patterns
        def pattern_1():
            return [" ".join(str(x) for x in range(1, i
+ 1)) for i in range(1, 6)]

        def pattern_2():
            return [" ".join(str(x) for x in range(1, i
+ 1)) for i in range(5, 0, -1)]
```

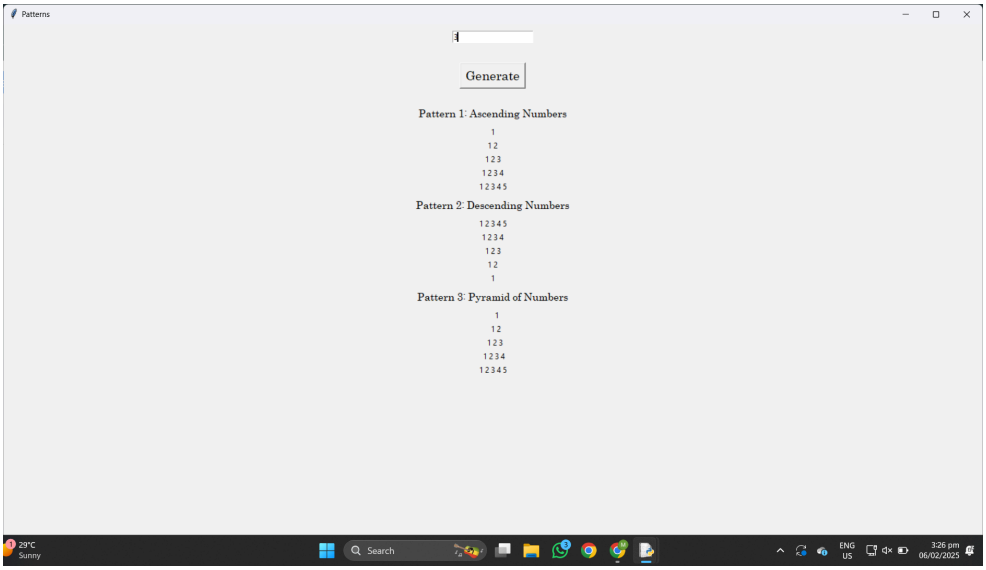
```
    def pattern_3():
        return [" " * (5 - i) + " ".join(str(x) for x
in range(1, i + 1)) for i in range(1, 6)]

    def pattern_4():
        return [" " * (5 - i) + " ".join(str(x) for x
in range(1, i + 1)) for i in range(5, 0, -1)]

    def pattern_5():
        return [" " * (5 - i) + " ".join(str(x) for x
in range(i, 0, -1)) for i in range(1, 6)]

    patterns.append(("Pattern 1: Ascending
Numbers", pattern_1))
    patterns.append(("Pattern 2: Descending
Numbers", pattern_2))
    patterns.append(("Pattern 3: Pyramid of
Numbers", pattern_3))
    patterns.append(("Pattern 4: Reverse
Pyramid", pattern_4))
    patterns.append(("Pattern 5:
Right-Aligned Numbers", pattern_5))

    # Display the required number of
patterns
    for i in range(num_patterns):
        title, pattern_func = patterns[i]
        Label(f, text=title, font=("Century",
12)).pack(pady=5)
```

```
            for line in pattern_func():
                Label(f, text=line).pack()

    except ValueError as e:
        Label(f, text=f"Error: {e}",
font=("Century", 12), fg="red").pack()

# Button to generate patterns
Button(f, text="Generate", font=("Century",
14), command=generate).pack(pady=20)

# Start the application loop
f.mainloop()
```

Output:



Enrollment no. 22FOTCA11071 6 -BCA-B **Name :** Mohmadhusen Khimani



*****