All Python Practical’s

Practical No.1:- Python Program for Pascal Triangle

```
n = int(input("Enter the number: "))
for i in range(n):
    print(" " * (n - i - 1), end="")
    print("*" * (2 * i + 1), end="")
    print(" " * (n - i - 1))
```

Output:-

```
    *
   ***
  *****
 *******
*********
```

Practical No.2:- Find out Roots of Quadratic Equations

```
from math import sqrt

print("Quadratic function: (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

if a == 0:
    print("The value of 'a' should not be zero in a quadratic equation.")
else:
    if r > 0:
        num_roots = 2
        x1 = ((-b) + sqrt(r)) / (2 * a)
        x2 = ((-b) - sqrt(r)) / (2 * a)
        print("There are 2 roots: %f and %f" % (x1, x2))
    elif r == 0:
        num_roots = 1
        x = (-b) / (2 * a)
        print("There is one root: ", x)
    else:
        num_roots = 0
        print("No roots, discriminant < 0.")
```

Output:-

```
Quadratic function: (a * x^2) + b*x + c
a:
5
b:
20
c:
10
There are 2 roots: -0.585786 and -3.414214
```

Practical No.3:- python program for display the Fibonacci series using generators

```python
n_terms = int(input("How many terms the user wants to print? "))
n_1 = 0
n_2 = 1
count = 0

if n_terms <= 0:
   print("Please enter a positive integer, the given number is not valid")
elif n_terms == 1:
   print("The Fibonacci sequence of the numbers up to", n_terms, ": ")
   print(n_1)
else:
   print("The Fibonacci sequence of the numbers is:")
   while count < n_terms:
      print(n_1, end=' ')
      nth = n_1 + n_2

      n_1 = n_2
      n_2 = nth
      count += 1
```

Output:-

```
How many terms the user wants to print?

5

The Fibonacci sequence of the numbers is:

0 1 1 2 3
```

Practical No.4:- To check the given number is Palindrome or Not

```
num = int(input("Enter a number: "))
temp = num
rev = 0
while(num > 0):
    dig = num % 10
    rev = rev * 10 + dig
    num = num // 10
if(temp == rev):
    print("The number is palindrome!")
else:
    print("Not a palindrome!")
```

Output:-

```
Enter a number:
121
The number is palindrome!
```

Practical No.5:- : To find the sum of digits of a given number

```
class Solution:
   def sum_of_digits(self, num):
      total = 0
      while num != 0:
         total += num % 10
         num //= 10
      return total

ob = Solution()

print(ob.sum_of_digits(512))
```

Output:-

```
8
```

Practical No.6:- Python program to remove the punctuations from a string.

```
punctuation = '!()-[]{};:\'"\\,<>./?@#$%^&*_~'
my_str = input("Enter a string: ")
no_punct = ""
for char in my_str:
   if char not in punctuation:
       no_punct += char
print(no_punct)
```

Output:-

```
Enter a string:
Python@#$123

Python123
```

Practical No.7:- Python program to implement the simple calculator

```
def add(P, Q):
   return P + Q

def subtract(P, Q):
   return P - Q

def multiply(P, Q):
   return P * Q

def divide(P, Q):
   if Q != 0:  # Check to prevent division by zero
      return P / Q
   else:
      return "Cannot divide by zero!"

print("Please select the operation.")
print("a. Add")
print("b. Subtract")
print("c. Multiply")
print("d. Divide")

choice = input("Please enter choice (a/ b/ c/ d): ")
```

```
num_1 = int(input("Please enter the first number: "))
num_2 = int(input("Please enter the second number: "))

if choice == 'a':
    print(num_1, " + ", num_2, " = ", add(num_1, num_2))

elif choice == 'b':
    print(num_1, " - ", num_2, " = ", subtract(num_1, num_2))

elif choice == 'c':
    print(num_1, " * ", num_2, " = ", multiply(num_1, num_2))

elif choice == 'd':
    if num_2 != 0:  # Check to ensure second number is not zero before dividing
        print(num_1, " / ", num_2, " = ", divide(num_1, num_2))
    else:
        print("Cannot divide by zero!")

else:
    print("This is an invalid input")
```

Output:-

```
Please select the operation.

a. Add

b. Subtract

c. Multiply

d. Divide


Please enter choice (a/ b/ c/ d):

A


Please enter the first number:

10


Please enter the second number:

20


10  +  20  =  30
```

Practical No.8:- Python program for reverse string.

```
def reverse_string(s):
   str1 = ""
   for i in s:
      str1 = i + str1
   return str1

original_str = "M.J.college,Jalgaon"
print("The original string is: ", original_str)
print("The reversed string is", reverse_string(original_str))
```

Output:-

```
The original string is:  M.J.college,Jalgaon
The reversed string is noaglaJ,egelloc.J.M
```

Practical No.9:- Python Program implementation of the Anonymous Function Lambda

```
x = lambda a, b, c : a + b + c
print(x(5, 6, 2))
```

Output:-

```
13
```

```
# Code to filter odd numbers from a given list

list_ = [34, 12, 64, 55, 75, 13, 63]
odd_list = list(filter(lambda num: (num % 2 != 0), list_))
print(odd_list)
```

Output:-

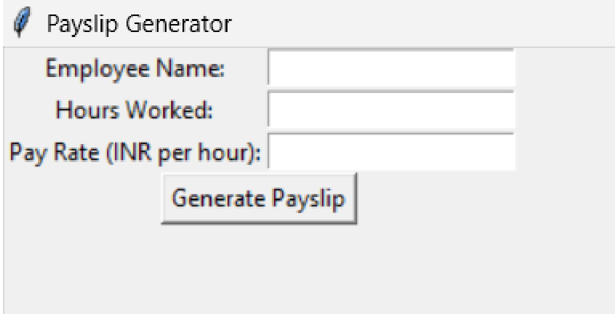
```
[55, 75, 13, 63]
```

Practical No.10:- Construct a GUI application to generate the employee pay slip.

```
import tkinter as tk

from tkinter import messagebox


def generate_payslip():

    try:

        # Get values from entries

        name = name_entry.get()

        hours_worked = float(hours_worked_entry.get())

        pay_rate = float(pay_rate_entry.get())

        gross_pay = hours_worked * pay_rate


        tax = gross_pay * 0.2  # Assume a flat 20% tax rate

        net_pay = gross_pay - tax


        payslip_message = f"""

        Employee Name: {name}

        Hours Worked: {hours_worked}

        Pay Rate: ₹{pay_rate:.2f} per hour

        Gross Pay: ₹{gross_pay:.2f}

        Tax Deducted (20%): ₹{tax:.2f}

        Net Pay: ₹{net_pay:.2f}
        """
        messagebox.showinfo("Payslip", payslip_message)
    except ValueError:
        messagebox.showerror(
            "Error", "Please enter numeric values for hours worked and
pay rate.")

root = tk.Tk()
root.title("Payslip Generator")

tk.Label(root, text="Employee Name:").grid(row=0, column=0)
name_entry = tk.Entry(root)
name_entry.grid(row=0, column=1)

tk.Label(root, text="Hours Worked:").grid(row=1, column=0)
hours_worked_entry = tk.Entry(root)
hours_worked_entry.grid(row=1, column=1)

tk.Label(root, text="Pay Rate (INR per hour):").grid(row=2, column=0)
pay_rate_entry = tk.Entry(root)
pay_rate_entry.grid(row=2, column=1)

generate_button = tk.Button(
    root, text="Generate Payslip", command=generate_payslip)
generate_button.grid(row=3, columnspan=2)
```

```
root.mainloop()
```

Output:-

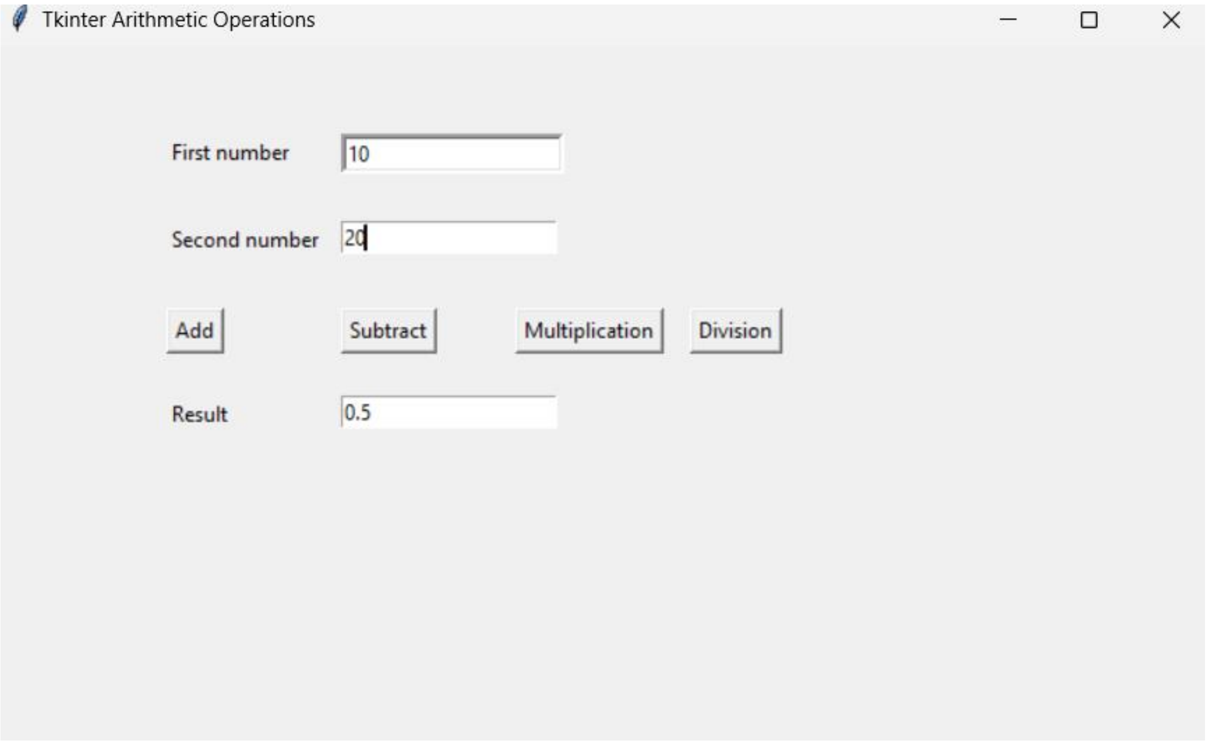


Practical No.11:- Construct a GUI application to perform the Arithmetic operations Read Input Values through input window and Display the result in Message Box.

```
from tkinter import *

class MyWindow:
    def __init__(self, win):
        self.lbl1 = Label(win, text='First number')
        self.lbl2 = Label(win, text='Second number')
        self.lbl3 = Label(win, text='Result')
        self.t1 = Entry(bd=3)
        self.t2 = Entry()
        self.t3 = Entry()
        self.lbl1.place(x=100, y=50)
        self.t1.place(x=200, y=50)
        self.lbl2.place(x=100, y=100)
        self.t2.place(x=200, y=100)

        self.b1 = Button(win, text='Add', command=self.add)
        self.b2 = Button(win, text='Subtract', command=self.sub)
        self.b3 = Button(win, text="Multiplication", command=self.mul)
        self.b4 = Button(win, text="Division", command=self.div)
        self.b1.place(x=100, y=150)
        self.b2.place(x=200, y=150)
        self.b3.place(x=300, y=150)
        self.b4.place(x=400, y=150)

        self.lbl3.place(x=100, y=200)
        self.t3.place(x=200, y=200)

    def add(self):
        self.t3.delete(0, 'end')
        num1 = int(self.t1.get())
        num2 = int(self.t2.get())
        result = num1 + num2
        self.t3.insert(END, str(result))

    def sub(self):
        self.t3.delete(0, 'end')
        num1 = int(self.t1.get())
        num2 = int(self.t2.get())
        result = num1 - num2
        self.t3.insert(END, str(result))

    def mul(self):
        self.t3.delete(0, 'end')
```

```
        num1 = int(self.t1.get())
        num2 = int(self.t2.get())
        result = num1 * num2
        self.t3.insert(END, str(result))

    def div(self):
        self.t3.delete(0, 'end')
        num1 = int(self.t1.get())
        num2 = int(self.t2.get())
        if num2 == 0:
            self.t3.insert(END, 'Error: Div by 0')
        else:
            result = num1 / num2
            self.t3.insert(END, str(result))


window = Tk()
mywin = MyWindow(window)
window.title('Tkinter Arithmetic Operations')
window.geometry("700x500+10+10")
window.mainloop()
```

Output:-



Tkinter Arithmetic Operations

First number 10

Second number 20

Result 0.5