```
# Constants
pi = 3.14159265359

# Input for Load and Depth
Q = float(input("Enter the value of Load in kN: "))
N = int(input("Number of data values of radial distance: "))
Z = float(input("Enter the Depth in meters: "))

# Array to store radial distances
r = []
# Input radial distances
for i in range(1, N + 1):
    Value_r = float(input(f"Enter radial distance {i} in meters: "))
    r.append(Value_r)

# Calculate and display stress for each radial distance
for Value_r in r:
    Stress = ((3 * Q) / (2 * pi * Z * Z)) * ((1 / (1 + ((Value_r / Z) ** 2))) ** 2.
    print(f"Stress at radial distance {Value_r} m: {Stress} kN/m^2")
```

```
Enter the value of Load in kN: 2500
Number of data values of radial distance: 5
Enter the Depth in meters: 6
Enter radial distance 1 in meters: 1
Enter radial distance 2 in meters: 2
Enter radial distance 3 in meters: 3
Enter radial distance 4 in meters: 4
Enter radial distance 5 in meters: 5
Stress at radial distance 1.0 m: 30.962130445358056 kN/m^2
Stress at radial distance 2.0 m: 25.479163627894877 kN/m^2
Stress at radial distance 3.0 m: 18.98033449112347 kN/m^2
Stress at radial distance 4.0 m: 13.22290223969301 kN/m^2
Stress at radial distance 5.0 m: 8.871775810212231 kN/m^2
```

```
# Input the values of load, vertical distance, and horizontal distance
Q = float(input("Enter the value of the given load (in kN): "))
z = float(input("Enter the vertical distance (in meters): "))
r = float(input("Enter the horizontal distance (in meters): "))

# Define the value of pi
pi = 3.14159265359

# Calculate the stress using Boussinesq's Theory
stress = (3 * Q * (1 / (1 + (r / z) ** 2)) ** 2.5) / (2 * pi * (z ** 2))

# Output the value of stress
print("The value of stress is", stress, "kN/m^2")
```

```
Enter the value of the given load (in kN): 2500
Enter the vertical distance (in meters): 6
Enter the horizontal distance (in meters): 5
The value of stress is 8.871775810212233 kN/m^2
```