```
# Stress When depth is constant
Q = float (input ("Enter the value of Load in kN: "))
N = int(input ("Number of data values of radial distance: "))
pi = 3.14159265359
Z = float(input ("Depth: "))
r = []
for i in range (1, N+1):
  print("Enter radial distance in m".format (i))
  Value_r = float(input () )
  r.append(Value_r)
  Stress = ((3*Q)/(2*pi*Z*Z) ) * (((1/(1+((Value_r/Z) **2))))**2.5)
  print("'Stress: ", Stress, "kN/m^2")
```

```
Enter the value of Load in kN: 2500
Number of data values of radial distance: 5
Depth: 6
Enter radial distance in m
1
'Stress:  30.962130445358056 kN/m^2
Enter radial distance in m
2
'Stress:  25.479163627894877 kN/m^2
Enter radial distance in m
3
'Stress:  18.98033449112347 kN/m^2
Enter radial distance in m
4
'Stress:  13.22290223969301 kN/m^2
Enter radial distance in m
5
'Stress:  8.871775810212231 kN/m^2
```

```
# Stress when Radius is Constant
Q = float (input("Enter the value of Load in kN: "))
M= int (input ("Number of data values of depth: "))
pi = 3.14159265359
r = float(input("Radial Distance: "))
Z = []
for j in range (1, M+1):
  print("Enter depth in z".format (j))
  Value_Z = float(input ())
  Z.append(Value_Z)
  Stress = ((3*Q)/(2*pi*Value_Z*Value_Z))*(((1/(1+((r/Value_Z)**2))))**2.5)
  print("'Stress: ", Stress, "kN/m^2")
```

```
Enter the value of Load in kN: 2500
Number of data values of depth: 6
Radial Distance: 5
Enter depth in z
1
'Stress:  0.346296438542730 23 kN/m^2
Enter depth in z
2
'Stress:  2.1085135063018074 kN/m^2
Enter depth in z
3
'Stress:  4.781320614736756 kN/m^2
Enter depth in z
4
'Stress:  7.0974399578803125 kN/m^2
Enter depth in z
5
'Stress:  8.440465463972316 kN/m^2
Enter depth in z
6
'Stress:  8.871775810212231 kN/m^2
```

```
# Calculating the stress by Boussineq's Theory
Q= int(input("Enter the value of given load :"))
z= int(input("Enter the distance of vertical stress :"))
r= int(input("Enter the distance ofhoriztal stress:"))
stress = ((3*Q*(1/(1+(r/z)**2)) **2.5))/(2*3.14*(z**2))
print("The value of stress is", stress)
```

```
Enter the value of given load :2500
Enter the distance of vertical stress :6
Enter the distance ofhoriztal stress:5
The value of stress is 8.876275703713446
```

```
# To find the downstream depth of open channel
# Given Data
Q= float (input("Enter the value of Discharge:4.8"))
T= int(input("Enter the value of top width:2"))
g=float(input("Enter the value of acceleration due to Gravity:9.81"))
y1 = float(input("enter the value of upstream depth:1.6"))
Z= float(input("Enter the Value of hump:0.1 "))
# Dicharge per meter width
q=Q/T
print ("The value of discharge per meter width is:", q)
# Area Calculation
A1= T*y1
print ("The value of upstream area is:", A1)
# Calculation of Froude Number
Fr1 = (Q*Q*T)/(g*A1* A1 *A1) * 0.5
print ("The value of Froude number is:", Fr1)
if Fr1>1:
    pass # Replace with desired code
    print("The flow is Super Critical Flow")
else:
    print("The flow is Sub Critical Flow")
#Upstream Energy
E1 = y1 + (Q*Q)/(2 *g*A1 *A1)
print ("The value of Energy at initial Section is:", E1)
# Downstream Energy
E2 = E1 -Z
print ("The value of downstream Energy E2 is:", E2)
# Critical Depth
# You need to define the values for 'a' and 'e' based on your problem.
# For example:
a = 1.0  # Replace with the actual value of 'a'
e = 1.0  # Replace with the actual value of 'e'
yc = (q*a/e)**0.3333
print ("The Value of critical depth is:", yc)
# Note: In Python, a tuple is defined using parenthesis. Using a comma creates a tuple.
# To perform multiplication, use the '*' operator.
Ec = 1.5*yc
print ("The value of critical Energy is", Ec)
if Ec>E2:
    print ("Chocking Condition")
else:
    print ("SAFE")
# Calculation of Zmax
# You need to define the value for 'E1' based on your problem.
# For example:
E1 = 1.8 # Replace with the actual value of 'E1'
Zmax =E1- Ec
print ("The value of maximum hump is:", Zmax)
```

```
Enter the value of Discharge:4.84.8
Enter the value of top width:22
Enter the value of acceleration due to Gravity:9.819.81
enter the value of upstream depth:1.61.6
Enter the Value of hump:0.1 0.1
The value of discharge per meter width is: 2.4
The value of upstream area is: 3.2
The value of Froude number is: 0.07167431192660548
The flow is Sub Critical Flow
The value of Energy at initial Section is: 1.714678899082569
The value of downstream Energy E2 is: 1.614678899082569
The Value of critical depth is: 1.3388268295597898
The value of critical Energy is 2.0082402443396847
Chocking Condition
The value of maximum hump is: -0.2082402443396847
```

```
# To find the downstream depth of open channel
# Given Data
Q= float(input("Enter the value of Discharge:15"))
B1 = float(input("Enter the value of width at upstream:3.5 "))
B2 = float(input("Enter the value of width at downstream:2.5 "))
g= float(input("Enter the value of acceleration due to Gravity:9.81"))
y1= float(input("enter the value of upstream depth:2"))
# Dicharge per meter width
q1=Q/B1
q2= Q/B2
print ("The value of discharge per meter width is:'", q1) # Changed q1 to q1
```

```
print ("The value of discharge per meter width is:", q2)
# Area Calculation
A1 = B1*yl # Changed y1 to yl as y1 was not defined
print ("The value of upstream area is:", A1)
# Calculation of Froude Number
Fr1 = ((Q*Q*B1)/(g*A1*A1*A1)) **0.5
print ("The value of Froude number is:", Fr1)
if Fr1>1:
    print("The flow is Super Critical Flow") # Indent this line to be part of the 'if' block
else:
    print("The flow is Sub Critical Flow")
# Upstream Energy
e = 1.602e-19 # Define 'e' here or import it from a library if it represents a physical constant
E1 = yl + (Q*0)/(2*e*A1*A1) # Use 'yl' instead of 'y1' if it's a different variable.
print ("The value of Energy at initial Section is:", E1) # Correct the variable name to 'E1'.
# (type alias) B2min: Any dition  # Remove or comment out this invalid line.
B2min = (27*Q*Q/(8*g*E1*E1*E1)) **0.5
print ("The value of minimum width to be kept to avoid Chocking is:", B2min)
if B2min > B2:
    print ("Chocking Condition") # Indent this line
else:
    print ("SAFE") # Indent this line to be part of the 'else' block
# Critical Depth
e = 0.8 # Or any other relevant value
yc = ((Q*Q)/(B2*B2*e)) **0.3333
print ("The Value of critical depth is: ", yc)
Ec = 1.5*yc
```

```
Enter the value of Discharge:1515
Enter the value of width at upstream:3.5 3.5
Enter the value of width at downstream:2.5 2.5
Enter the value of acceleration due to Gravity:9.819.81
enter the value of upstream depth:22
The value of discharge per meter width is:' 4.285714285714286
The value of discharge per meter width is: 6.0
The value of upstream area is: 7.0
The value of Froude number is: 0.4837753296275688
The flow is Sub Critical Flow
The value of Energy at initial Section is: 2.0
The value of minimum width to be kept to avoid Chocking is: 3.110632107802487
Chocking Condition
The Value of critical depth is:  3.5564420033791078
The value of critical Energy is 5.334663005068662
```

```
#Design of Efficient Channel Section
Q= float(input("Enter the value of Discharge:100"))
n=float(input("Enter the value of Rugosity coefficient:0.015"))
So= float (input("Enter the value of bed slope:0.0004"))
g= float(input("Enter the value of acceleration due to Gravity:9.81"))
#Manning's Formula
#Q = (AR^2/3 S^1/2)/n
yn = (Q*n*50* 1.591)/(1.732) * (3/8)
print ("The Value of yn is", yn)
#To encounter the effect of free board
yn1= 1.1*yn
print ("The Value of yn1 is", yn1)
# Cross Sectional Area
A = 1.732 * yn * yn1
print ("The cross sectional Area is:", A)
# Top Width
T= 4* yn/1.732
print ("The value of top Width is:", T)
# Bottom Width
B= 2 * yn/1.732
print ("The value of Bottom Width is'", B)
Fr= ((Q*Q*T)/(g*A*A*A)) * 0.5 # Fixed the Fr calculation
print ("The value of Froude number is:", Fr)
if Fr>1:
    print("The flow is Super Critical Flow") # Indented this line
else:
    print("The flow is Sub Critical Flow")
```

```
Enter the value of Discharge:100100
Enter the value of Rugosity coefficient:0.0150.015
Enter the value of bed slope:0.00040.0004
Enter the value of acceleration due to Gravity:9.819.81
The Value of yn is 25.83537817551963
The Value of yn1 is 28.418915993071593
```