Python Solution for Mechanics Problems

In [1]:
```
'''
    Type: Bananna
    Program no.: 1
    Question no.: 1
    Problem no. as per Question PDF: 2
'''
from math import acos, degrees
F1 = float(input("Enter the magnitude of F1 (in N): "))
F2 = float(input("Enter the magnitude of F2 (in N): "))
R = float(input("Enter the resultant force (in N): "))
cos_theta = (R**2 - F1**2 - F2**2) / (2 * F1 * F2)
theta = degrees(acos(cos_theta))
print(f"The angle between the forces is: {theta:.4f} degrees")
```

```
The angle between the forces is: 51.3178 degrees
```

In [16]:
```
'''
    Type: Bananna
    Program no.: 2
    Question no.: 2
    Problem no. as per Question PDF: 5
'''
from math import sin, cos, tan, radians
tension = float(input('Enter The Tension T: '))

#To Find Tension T in Vector Form
print(f'The Tension T in vector Form: {'{:.2f}'.format(tension * sin(radians(20)

#To Calculate Moment at Mb and Mo
M_b = float('{:.2f}'.format((tension * sin(radians(50)) * (0.298))))
print(f'Moment at B: {M_b} Nm')
print(f'The Moment at O: {'{:.2f}'.format(M_b + (tension * sin(radians(50)) * (0
```

```
The Tension T in vector Form: 266.78i - 732.96j N
Moment at B: 178.06
The Moment at O: 253.94
```

In [3]:
```
'''
    Type: Orange
    Program no.: 3
    Question no.: 3
    Problem no. as per Question PDF: 8
'''

mass = float(input("Enter the mass of the load (in kg): "))
num_segments = int(input("Enter the number of cable segments supporting the load

g = 9.81  # Acceleration due to gravity in m/s^2

total_tension = mass * g  # Total tension in the system
tension_per_segment = total_tension / num_segments  # Tension per segment

print(f"Total tension in the cable system (T): {total_tension:.2f} N")
print(f"Tension in each segment (T/{num_segments}): {tension_per_segment:.2f} N"
```

```
Total tension in the cable system (T): 981.00 N
Tension in each segment (T/4): 245.25 N
```

In [6]:
```
'''
    Type: Orange
    Program no.: 4
    Question no.: 4
    Problem no. as per Question PDF: 9
'''
from math import sin, cos, tan, radians

weight = float(input("Enter the weight of the object (kg): "))
angle_AC = float(input("Enter the angle between AC and vertical (degrees): "))
angle_BC = float(input("Enter the angle between BC and vertical (degrees): "))

g = 9.81

T_CD = weight * g
print(f"Tension in CD (T_CD): {T_CD:.2f} N")
T_AC = T_CD * cos(radians(angle_AC))
print(f"Tension in AC (T_AC): {T_AC:.2f} N")
T_BC = T_CD * cos(radians(angle_BC))
print(f"Tension in BC (T_BC): {T_BC:.2f} N")
```

```
Tension in CD (T_CD): 294.30 N
Tension in AC (T_AC): 147.15 N
Tension in BC (T_BC): 208.10 N
```

In [14]:
```
'''
    Type: Jackkfruit
    Program no.: 5
    Question no.: 5
    Problem no. as per Question PDF: 18
'''
from math import sin, cos, tan, radians
man_weight = float(input('Enter the weigth of the Man: '))
cart_weight = float(input('Enter the weigth of the Cart: '))

theta = 16.667

# To find T
T = float('{:.2f}'.format((cart_weight * sin(radians(theta))) / cos(radians(40))

#To Find N
N = float('{:.2f}'.format(man_weight * sin(radians(theta)) + T * cos(radians(40)

#To Find F
F  = float('{:.2f}'.format(man_weight * cos(radians(theta)) + T * cos(radians(40

#To Find Mu
print(f"The Co-efficient of Friction is: {'{:.3f}'.format(N/F)}")

#To Find Distance
dist = '{:.2f}'.format((T * 175 * sin(radians(40 + theta)) + T * 850 * cos(radia
print(f"The Minimum Distance is: {dist} m")
```

```
The Co-efficient of Friction is: 0.398
The Minimum Distance is: 126.02
```

In [17]:
```
'''
    Type: Jackkfruit
    Program no.: 6
    Question no.: 6
```

```
    Problem no. as per Question PDF: 19
'''
from math import sin, cos, tan, radians
painter_weight = float(input('Enter mass of the painter: '))
ladder_weight = float(input('Enter the mass of the ladder: '))
ladder_height = float(input('Enter Ladder Height: '))
static_friction = float(input('Enter the Co-Efficient of static friction: '))


#To Find Na
N_a = painter_weight * 9.81 + ladder_weight * 9.81

#To Find X
X = (N_a * 1.5 - ladder_weight * 9.81 * 0.75 - static_friction * N_a * 3.71 ) /
X = float('{:.3f}'.format(X))
D = 1.5 - X
cosine_of_ladder = ladder_height / 1.5

#To Find Distance S
S = D * cosine_of_ladder

print(f'The Maximum Distance that Painter Can Climb without Sliping is: {'{:.2f}
```

```
The Maximum Distance that Painter Can Climb without Sliping is: 2.55 m
```