SFD and BMD on simple suported beam For a single point load

```
import numpy as np
import matplotlib.pyplot as plt
```

```
P = float(input('load = '))
u1 = input('load unit = ')
L = float(input('Length of the beam = '))
u2 = input('length unit = ')
a = float(input('Distance of Point load from left end = '))
```

```
load = 345
load unit = KN
Length of the beam = 15
length unit = m
Distance of Point load from left end = 23
```

```
b = L - a
R1 = P*b/L
R2 = P - R1
R1 = round(R1, 3)
R2 = round(R2, 3)
print(f'''
As per the static equilibrium, net moment sum at either end is zero,
hence Reaction R1 = P*b/L = {R1} {u1},
Also Net sum of vertical forces is zero,
hence R1+R2 = P, R2 = P - R1 = {R2} {u1}.
''')
```

```
As per the static equilibrium, net moment sum at either end is zero,
hence Reaction R1 = P*b/L = -184.0 KN,
Also Net sum of vertical forces is zero,
hence R1+R2 = P, R2 = P - R1 = 529.0 KN.
```

```
l = np.linspace(0, L, 1000)
X = []
SF = []
M = []
maxBM= float()
for x in l:
    if x <= a:
        m = R1*x
        sf = R1
    elif x > a:
        m = R1*x - P*(x-a)
```

```
        sf = -R2
    M.append(m)
    X.append(x)
    SF.append(sf)
```

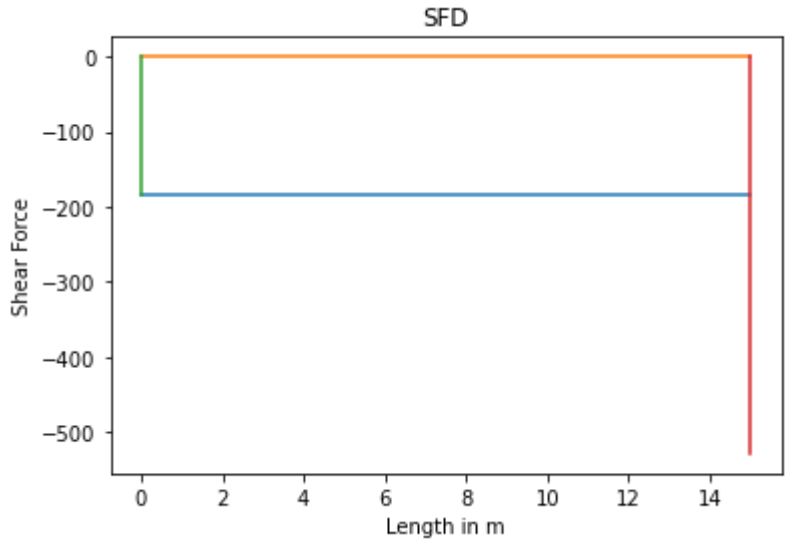
```
print(f'''
Shear Force at x (x<{a}), Vx = R1 ={R1} {u1}
             at x (x>{a}), SF = R1 - P = {R1} - {P} = -{R1-P} {u1}

Bending Moment at x (x<{a}), Mx = R1*x = {R1}*x
               at x (x>={a}), Mx = R1*x - P*(x-{a})
                                 = {R1}x - {P}(x-{a}) = -{R2}x + {P*a}''')
```

```
Shear Force at x (x<23.0), Vx = R1 =-184.0 KN
            at x (x>23.0), SF = R1 - P = -184.0 - 345.0 = --529.0 KN

Bending Moment at x (x<23.0), Mx = R1*x = -184.0*x
              at x (x>=23.0), Mx = R1*x - P*(x-23.0)
                              = -184.0x - 345.0(x-23.0) = -529.0x + 7935.0
```

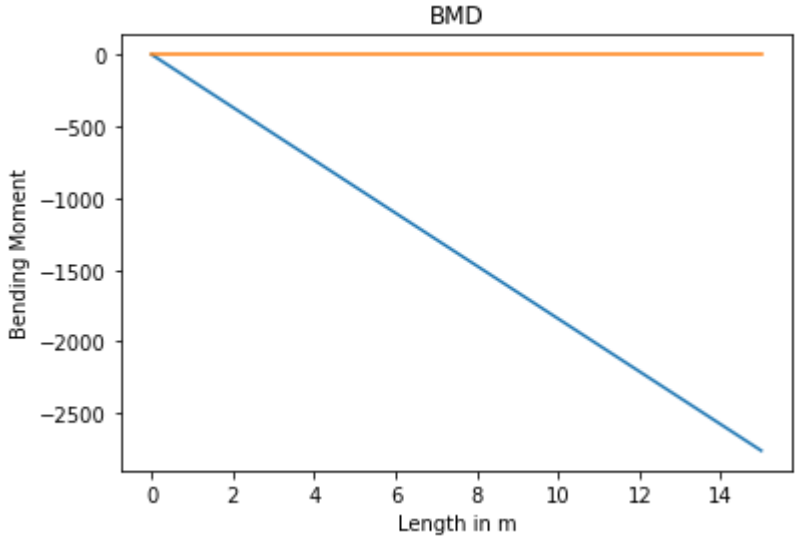
```
max_SF = 0
for k in SF:
    if max_SF < k:
        max_SF = k


print(f'Maximum Shear Force Vmax = {max_SF} {u1}')


for k in M:
    if maxBM < k:
        maxBM = k
print(f'maximum BM, Mmax = {round(maxBM, 3)} {u1}{u2}')
Mx = float()

for x in l:
   if x<a:
       Mx = R1*x
       if maxBM == Mx:
          print(f'maximum BM at x = {round(x,3)} {u2}')
   elif x>=a:
        Mx = R1*x - P*(x- a)
        if maxBM == Mx:
          print(f'maximum BM at x = {round(x,3)} {u2}')
```

```
Maximum Shear Force Vmax = 0 KN
maximum BM, Mmax = 0.0 KNm
maximum BM at x = 0.0 m
```

```
plt.plot(X, SF)
plt.plot([0, L], [0, 0])
plt.plot([0, 0], [0, R1], [L, L], [0, -R2])
plt.title("SFD")
plt.xlabel("Length in m")
plt.ylabel("Shear Force")
plt.show()
```




```
plt.plot(X, M)
plt.plot([0, L], [0, 0])
plt.title("BMD")
plt.xlabel("Length in m")
plt.ylabel("Bending Moment")
plt.show()
```




```
import jovian
jovian.commit
```

```
<function jovian.utils.commit.commit(message=None, files=[], outputs=[],
environment=None, privacy='auto', filename=None, project=None, new_project=None,
git_commit=False, git_message='auto', require_write_access=False, **kwargs)>
```