```
# Lets us consider we have Forward speed module -1,2,3,4,5 and backward module-6, now consider them as vectors and arrange on matrix
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

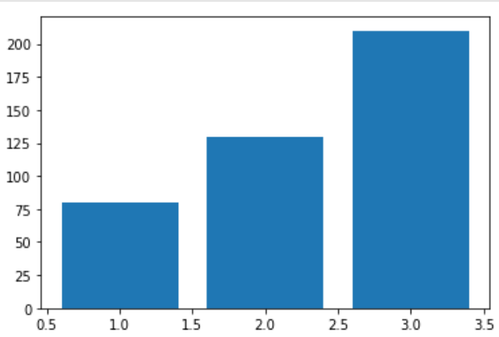
In [ ]:

```
arr1=np.array(([1,1],[1,2],[1,3],[1,4],[1,5]))
```

In [1]:
```
import numpy as np
```

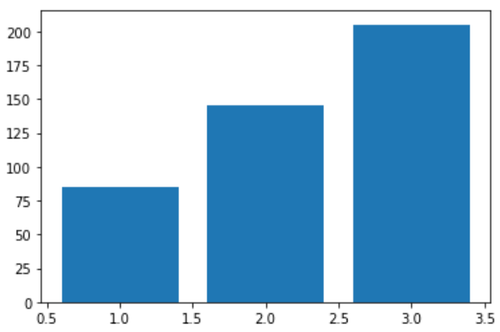
In [4]:
```
# now we need to calculate 5 paramaters here because in arr1 is 5dimensions,
```

In [5]:
```
a=[1,2,3]
b=[80,130,210]
plt.bar(a,b)
plt.show()
```



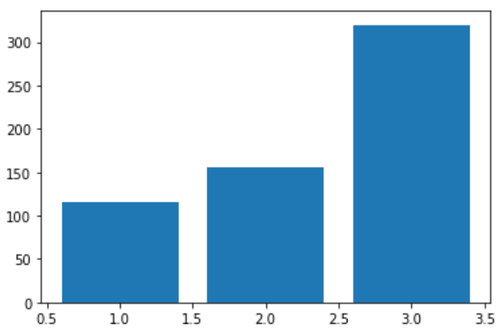

In [ ]:
```
# hence we have plotted first parameter b=[80,130,210] on graph for data visualisation, now consider second parameter,
```

In [6]:
```
c=[1,2,3]
d=[85,145,205]
plt.bar(c,d)
plt.show()
```



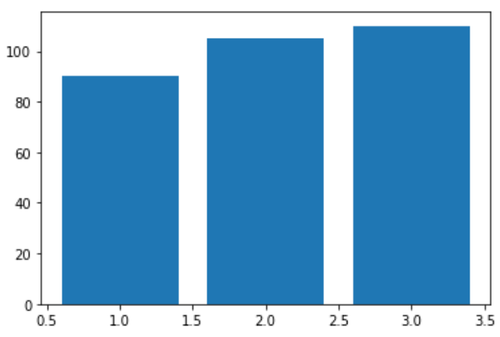

In [ ]:
```
# hence we have plotted second parameter d=[85,145,205] on graph for data visualisation to machine now consider third parameter
```

In [7]:
```
e=[1,2,3]
f=[115,155,320]
plt.bar(e,f)
plt.show()
```



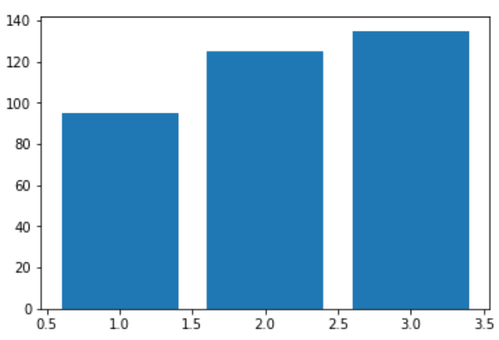

In [ ]:
```
# hence we have plotted 3rd parameter f=[115,155,320] on graph for data visualisation, now consider 4th parameter
```

In [8]:
```
g=[1,2,3]
h=[90,105,110]
plt.bar(g,h)
plt.show()
```




In [ ]:
```
# hence we have calculated forth parameter, now consider 5th parameter
```

In [9]:
```
i=[1,2,3]
j=[95,125,135]
plt.bar(i,j)
plt.show()
```




In [ ]:
```
# hence we calculated 5 parameters, now apply machine prediction formula - data matrix * parameters, (before arrange all parameters on matrix)
```

In [10]:
```
arr2=np.array(([80,130,210],[85,145,205],[115,155,320],[90,105,110],[95,125,135]))
```

In [11]:
```
arr2
```

Out[11]:
```
array([[ 80, 130, 210],
       [ 85, 145, 205],
       [115, 155, 320],
       [ 90, 105, 110],
       [ 95, 125, 135]])
```

In [ ]:
```

```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"



Loading [MathJax]/extensions/Safe.js

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [2]:
```
import sympy as smp
from sympy import *
```

In [3]:
```
x,y = smp.symbols('x y')
```

In [4]:
```
x**2+y
```

Out[4]: $\displaystyle x^{2} + y$

In [5]:
```
smp.sin(x)
```

Out[5]: $\displaystyle \sin{\left(x \right)}$

In [6]:
```
smp.diff((x),x)
```

Out[6]: $\displaystyle 1$

In [9]:
```
smp.diff(((1+smp.sin(x)))/(1-smp.cos(x))**2,x)
```

Out[9]: $\displaystyle \frac{\cos{\left(x \right)}}{\left(1 - \cos{\left(x \right)}\right)^{2}} - \frac{2 \left(\sin{\left(x \right)} + 1\right) \sin{\left(x \right)}}{\left(1 - \cos{\left(x \right)}\right)^{3}}$

In [10]:
```
smp.integrate(smp.csc(x)*smp.cot(x),x)
```

Out[10]: $\displaystyle - \frac{1}{\sin{\left(x \right)}}$

In [11]:
```
smp.integrate(4*smp.sec(3*x)*smp.tan(3*x),x)
```

Out[11]: $\displaystyle \frac{4}{3 \cos{\left(3 x \right)}}$

In [ ]:
```
2/smp.sqrt(1-x**2)-1/x**smp.Rational(1,)
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Let consider rate of change of position of Cam with respect to seconds of time during
```

In [4]:
```
import sympy as smp
from sympy import *
```

In [5]:
```
x,y = smp.symbols('x y')
```

In [6]:
```
smp.diff(x)
```

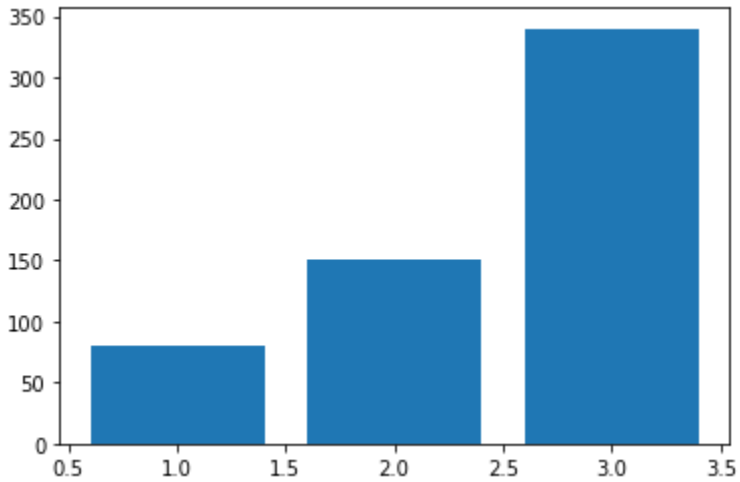
Out[6]: $\displaystyle 1$

In [ ]:
```
# let us calculate atleast 3 hypothesis for machine learning, h=[h1+h2+h3.....nth],
```

In [7]:
```
import matplotlib.pyplot as plt
```

In [10]:
```
a=[1,2,3]
b=[80,150,340]
plt.bar(a,b)
plt.show
```

Out[10]:
```
<function  matplotlib.pyplot.show(close=None,  block=None)>
```

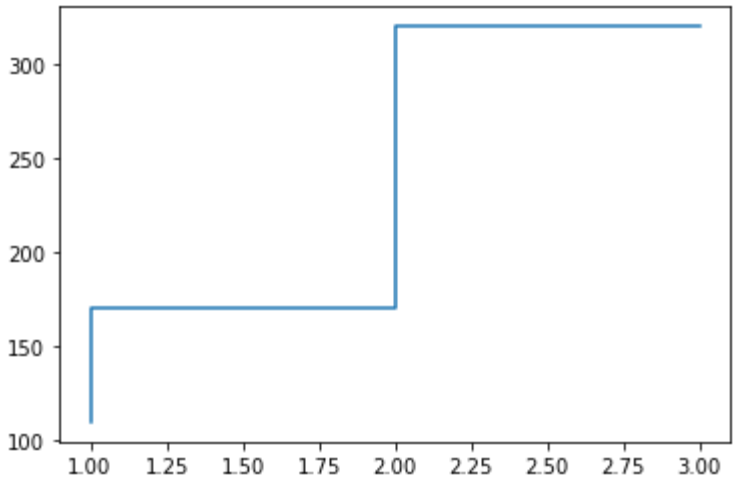



In [ ]:
```
# hence we got first set of parameters=80,150,340 plotted successfully on machine for da
```

In [13]:
```
c=[1,2,3]
d=[110,170,320]
plt.step(c,d)
plt.show()
```

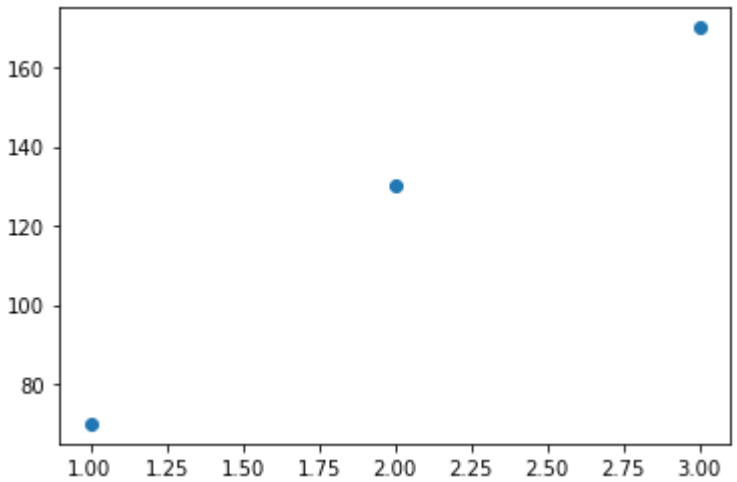
# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"




In [ ]:
```
# hence we got second set of parameters successfully plotted on machine for data visiual
```

In [12]:
```
e=[1,2,3]
f=[70,130,170]
plt.scatter(e,f)
plt.show()
```




In [ ]:
```
#hence we have got third parameter successfully plotted on machine  for data visualisati
```

In [ ]:
```
# arrange set of parameters on Matrix for machine predictions
```

In [14]:
```
import numpy as np
```

In [17]:
```
arr1=np.array(([80,150,340],[110,170,320],[70,130,170]))
```

In [18]:
```
arr1
```

Out[18]:
```
array([[ 80, 150, 340],
       [110, 170, 320],
       [ 70, 130, 170]])
```

In [ ]:
```
# now apply machine prediction formula= Data Matrix * Parameters
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [19]:
```
x*arr1
```

Out[19]:
```
array([[80*x, 150*x, 340*x],
       [110*x, 170*x, 320*x],
       [70*x, 130*x, 170*x]], dtype=object)
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Let us consider engine firing order be = 1342, before plug in the equation to machine first arrange data Matrix and parameters.
```

In [1]:
```
import numpy as np
```

In [2]:
```
arr1=np.array(([1,1],[1,3],[1,4],[1,2]))
```

In [3]:
```
arr1
```

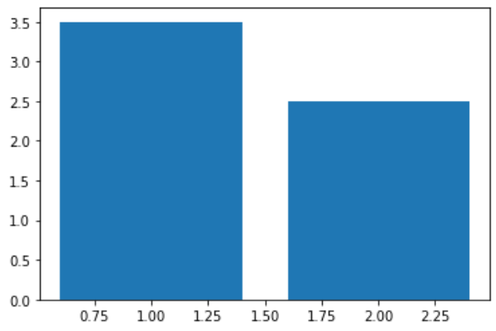
Out[3]:
```
array([[1, 1],
       [1, 3],
       [1, 4],
       [1, 2]])
```

In [ ]:
```
# as above we have arranged data matrix, now calculate atleast 3 parameters for machine learning, h=[h1+h2+h3.....nth]
```

In [ ]:
```
# first calculate h1, plot the graph
```

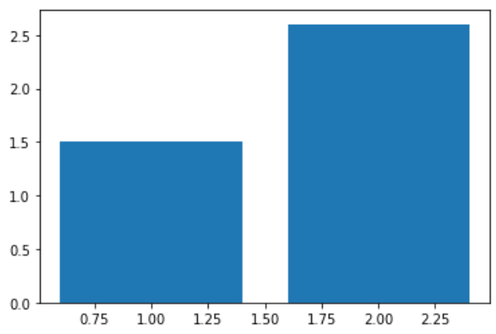
In [4]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
a=[1,2]
b=[3.5,2.5]
plt.bar(a,b)
plt.show()
```



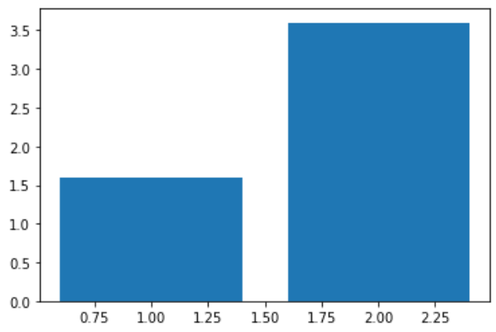

In [ ]:
```
# hence, we got firtst parameters (b=3.5,2.5) plotted on the graph for machine data visualisation, now plot second parameter.
```

In [9]:
```
c=[1,2]
d=[1.5,2.6]
plt.bar(c,d)
plt.show()
```




In [ ]:
```
# hence we have plotted second parameter d=1.2,2.6 on graph for machine learning, now consider third parameter
```

In [10]:
```
e=[1,2]
f=[1.6,3.6]
plt.bar(e,f)
plt.show()
```




In [ ]:
```
# hence we have got third parameter f=[1.6,3.6] plotted on graph for data visualisation, now arrange all the 3 parameters on matrix
```

In [11]:
```
arr2=np.array(([3.5,2.5],[1.6,2.6],[1.6,3.6]))
```

In [12]:
```
arr2
```

Out[12]:
```
array([[3.5, 2.5],
       [1.6, 2.6],
       [1.6, 3.6]])
```

In [ ]:
```
# now apply machine prediction formula = data Matrix * all three hypothesis(paramters) and the equation would be plug in to machine
```

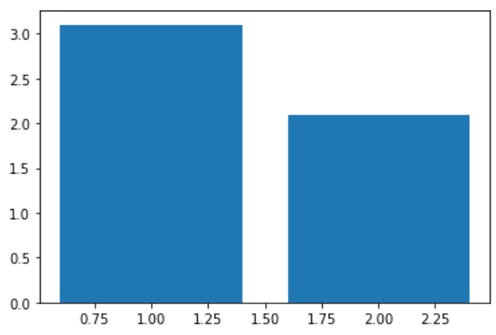
In [13]:
```
arr1*arr2
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [13], in <cell line: 1>()
----> 1 arr1*arr2

ValueError: operands could not be broadcast together with shapes (4,2) (3,2)
```

In [ ]:
```
# dimenison 4,2 & 3,2 hence we need to correct equal dimesions matrix so now consider 4th parameter,
```

In [14]:
```
g=[1,2]
h=[3.1,2.1]
plt.bar(g,h)
plt.show()
```




In [ ]:
```
# hence we got forth parameter g=3.1,2.1, now arrange matrix once again
```

In [15]:
```
arr2=np.array(([3.5,2.6],[1.6,2.6],[1.6,3.6],[3.1,2.1]))
```

In [16]:
```
arr2
```

Out[16]:
```
array([[3.5, 2.6],
       [1.6, 2.6],
       [1.6, 3.6],
       [3.1, 2.1]])
```

In [ ]:
```
# now finally apply machine prediction formula=data Matrix * paramters
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [17]:
```
arr1*arr2
```

Out[17]:
```
array([[ 3.5,  2.6],
       [ 1.6,  7.8],
       [ 1.6, 14.4],
       [ 3.1,  4.2]])
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Let us consider rate of change in position of flywheel with respect to seconds of time (t) be = pow(x,n) (as the rate of change in position is nth  times)
```

In [1]:
```
import sympy as smp
from sympy import *
```

In [5]:
```
x,n = smp.symbols('x n')
```

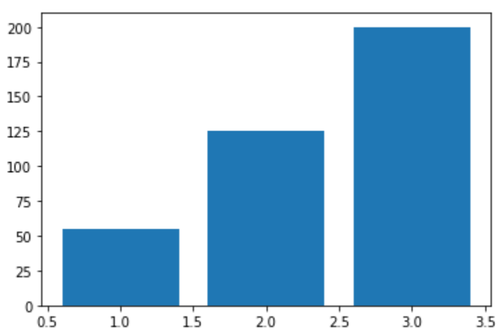
In [9]:
```
smp.diff(pow(x,3)),x
```

Out[9]:
```
(3*x**2, x)
```

In [ ]:
```
# Now lets consider atleast 3 hypothesis for machine learning, h=[h1+h2+h3.....nth], now plot the graph,
```

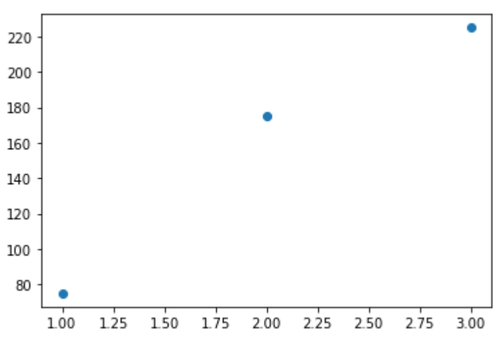
In [10]:
```
import matplotlib.pyplot as plt
```

In [11]:
```
a=[1,2,3]
b=[55,125,200]
plt.bar(a,b)
plt.show()
```



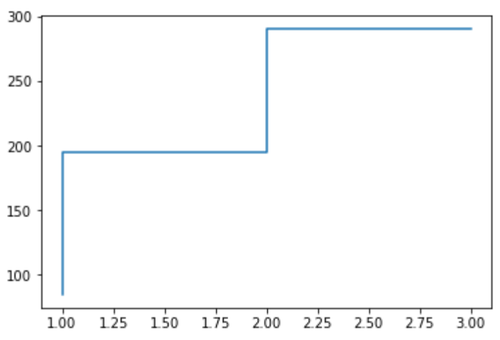

In [ ]:
```
# hence we have successfully plotted first parameter on machine for data visualisation, now consider second parameter,
```

In [12]:
```
c=[1,2,3]
d=[75,175,225]
plt.scatter(c,d)
plt.show()
```




In [ ]:
```
# hence we have successfully plotted second parameter on machine for data visualisation, now let us consider third parameter.
```

In [13]:
```
e=[1,2,3]
f=[85,195,290]
plt.step(e,f)
plt.show()
```




In [ ]:
```
# hence we have successfully plotted third parameter on machine for data visualisation, now Dendrite funciton= Data Matrix * Parametes
```

In [ ]:
```
# arrange all set of parameters on Matrix
```

In [14]:
```
import numpy as np
```

In [15]:
```
arr1=np.array(([55,125,200],[75,175,225],[85,195,290]))
```

In [16]:
```
arr1
```

Out[16]:
```
array([[ 55, 125, 200],
       [ 75, 175, 225],
       [ 85, 195, 290]])
```

In [ ]:
```
# now data matrix * parameters
```

In [17]:

Out[17]:

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

```
smp.diff(pow(x,3)),x*arr1
```

```
(3*x**2
,
 a
        r
        r
        a
        y
        (
        [
        [
        5
        5
        *
        x
        ,

        1
        2
        5
        *
        x
        ,

        2
        0
        0
        *
        x
        ]
        ,

        [
        7
        5
        *
        x
        ,

        1
        7
        5
        *
        x
        ,
        2
        2
        5
        *
        x
        ]
        ,
        [
        8
        5
        *
        x
        ,

        1
        9
        5
        *
        x
        ,

        2
        9
        0
        *
        x
        ]
        ]
        ,

        d
        t
        y
        p
        e
        =
        o
        b
        j
        e
        c
        t
        )
        )
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Neural Schema"

```
# Lets us consider we have Forward speed module -1,2,3,4,5 and backward module-6, now consider them as vectors and arrange on matrix
```


Prem Kumar

Charter Automobile Engineer Bihar"


In [ ]:

In [1]:
```
import numpy as np
```

In [2]:
```
arr1=np.array(([1,1],[1,2],[1,3],[1,4],[1,5]))
```

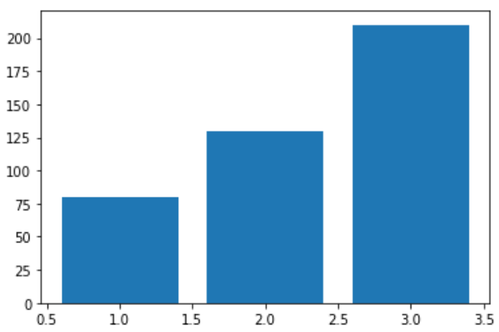
In [3]:
```
arr1
```

Out[3]:
```
array([[1, 1],
       [1, 2],
       [1, 3],
       [1, 4],
       [1, 5]])
```

In [ ]:
```
# now we need to calculate 5 paramaters here because in arr1 is 5dimensions,
```

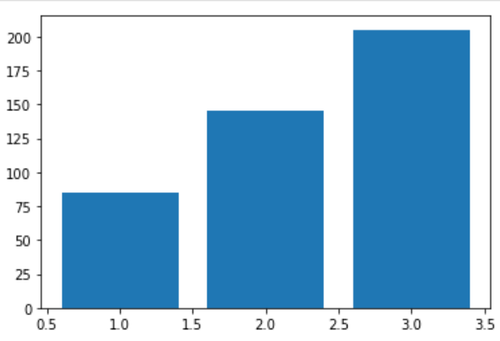
In [4]:
```
import matplotlib.pyplot as plt
```

In [5]:
```
a=[1,2,3]
b=[80,130,210]
plt.bar(a,b)
plt.show()
```



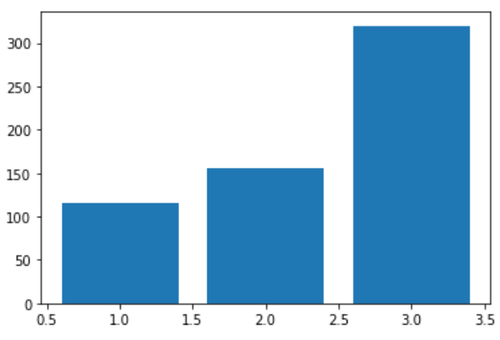

In [ ]:
```
# hence we have plotted first parameter b=[80,130,210] on graph for data visualisation, now consider second parameter,
```

In [6]:
```
c=[1,2,3]
d=[85,145,205]
plt.bar(c,d)
plt.show()
```



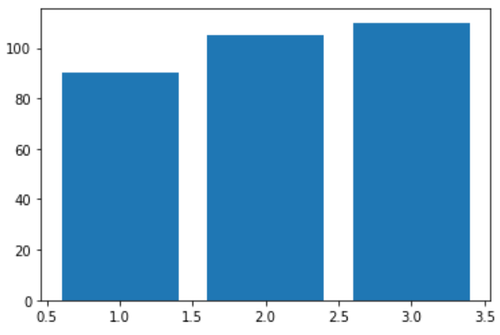

In [ ]:
```
# hence we have plotted second parameter d=[85,145,205] on graph for data visualisation to machine now consider third parameter
```

In [7]:
```
e=[1,2,3]
f=[115,155,320]
plt.bar(e,f)
plt.show()
```



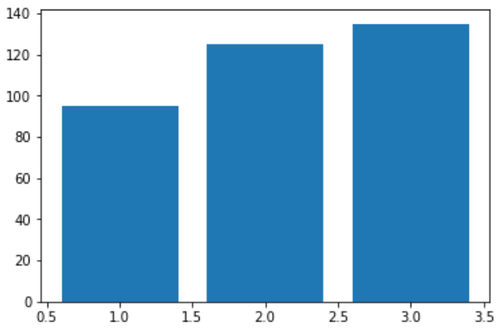

In [ ]:
```
# hence we have plotted 3rd parameter f=[115,155,320] on graph for data visualisation, now consider 4th parameter
```

In [8]:
```
g=[1,2,3]
h=[90,105,110]
plt.bar(g,h)
plt.show()
```




In [ ]:
```
# hence we have calculated forth parameter, now consider 5th parameter
```

In [9]:
```
i=[1,2,3]
j=[95,125,135]
plt.bar(i,j)
plt.show()
```




In [ ]:
```
# hence we calculated 5 parameters, now apply machine prediction formula - data matrix * parameters, (before arrange all parameters on matrix)
```

In [10]:
```
arr2=np.array(([80,130,210],[85,145,205],[115,155,320],[90,105,110],[95,125,135]))
```

In [11]:

Out[11]:

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

```
arr2
```

```
array([[ 80, 130, 210],
       [ 85, 145, 205],
       [111, 155, 320],
       [ 90, 105, 110],
       [ 95, 125, 135]])
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [25]:
```
Sensor = {"True":1,"False":0}

If [Sensor==True]:
     print("Jam Alert")
Else:
    print("Jam clear")
```

In [ ]:
```
  Input In [25]
    If [Sensor==True]:
                     ^
SyntaxError: invalid syntax
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [2]:
```
import matplotlib.pyplot as plt
```

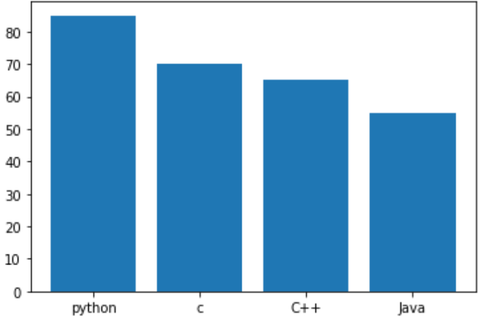
In [3]:
```
x=["python","c","C++","Java"]
y=[85,70,65,55]
plt.bar(x,y)
plt.show()
```

In [ ]:
```

```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"




In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Let us consider rate of change in position of Reverse gear with respect to time(t) be = pow(x,1/2)
```

In [1]:
```
import sympy as smp
from sympy import *
```

In [4]:
```
x,y = smp.symbols('x y')
```

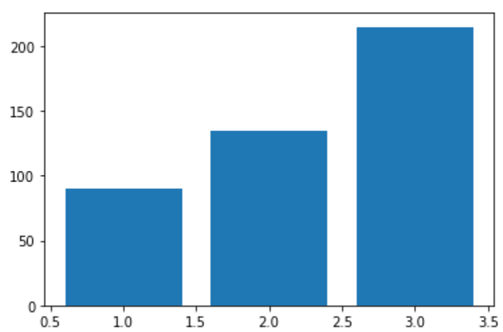
In [12]:
```
smp.diff(pow(x,1/2)),x
```

Out[12]:
```
(0.5/x**0.5, x)
```

In [ ]:
```
# now let us calculate atleast 3 parameters for machine learning, h=[h1+h2+h3.....nth] and plot the graph
```

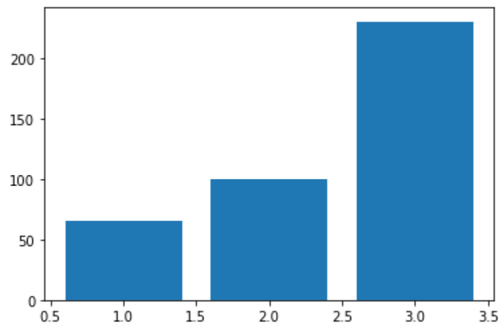
In [13]:
```
import matplotlib.pyplot as plt
```

In [14]:
```
a=[1,2,3]
b=[90,135,215]
plt.bar(a,b)
plt.show()
```



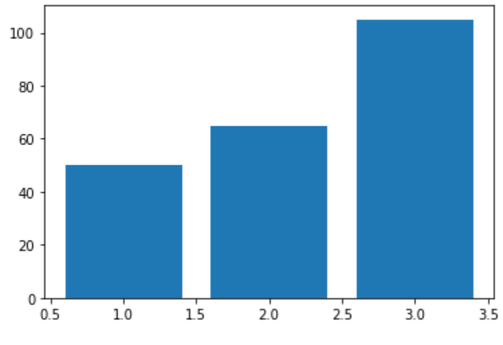

In [ ]:
```
# hence we have successfully plotted parameters on machine for data visualisation, now plot third parameter on machine
```

In [16]:
```
c=[1,2,3]
d=[65,100,230]
plt.bar(c,d)
plt.show()
```




In [ ]:
```
# hence we have plotted second parameter on machine for data visualisation, now plot third parameter.
```

In [17]:
```
e=[1,2,3]
f=[50,65,105]
plt.bar(e,f)
plt.show()
```




In [ ]:
```
# hence we have plotted third parameter on machine now plug in machine function for prediction= Data Matrix * parameters
```

In [ ]:
```
# arrange oll parameters on Matrix
```

In [18]:
```
import numpy as np
```

In [19]:
```
arr1=np.array(([90,135,215],[55,100,230],[50,65,105]))
```

In [20]:
```
arr1
```

Out[20]:
```
array([[ 90, 135, 215],
       [ 55, 100, 230],
       [ 50,  65, 105]])
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Data Matrix * Parameters Matrix
```

In [ ]:
```
smp.diff(pow(x,1/2))
```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

# " Car ECU Neural Schema"

```
# Lets us consider we have Forward speed module -1,2,3,4,5 and backward module-6, now consider them as vectors and arrange on matrix
```


Prem Kumar

Charter Automobile Engineer Bihar"


In [ ]:

In [1]:
```
import numpy as np
```

In [2]:
```
arr1=np.array(([1,1],[1,2],[1,3],[1,4],[1,5]))
```

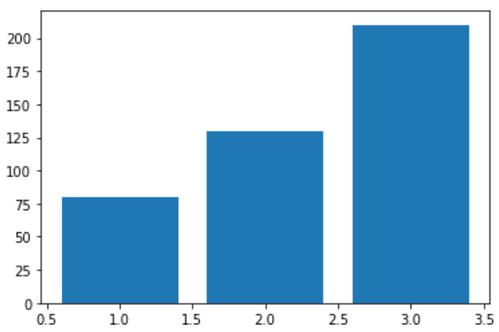
In [3]:
```
arr1
```

Out[3]:
```
array([[1, 1],
       [1, 2],
       [1, 3],
       [1, 4],
       [1, 5]])
```

In [ ]:
```
# now we need to calculate 5 paramaters here because in arr1 is 5dimensions,
```

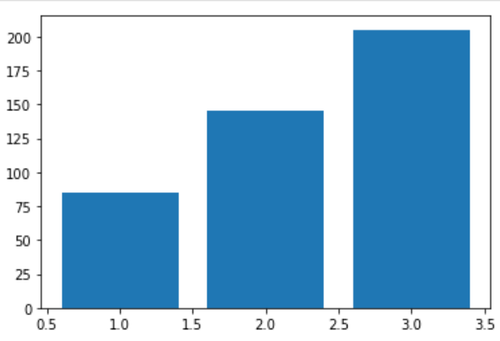
In [4]:
```
import matplotlib.pyplot as plt
```

In [5]:
```
a=[1,2,3]
b=[80,130,210]
plt.bar(a,b)
plt.show()
```



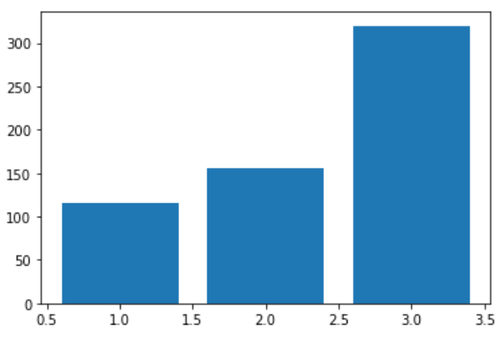

In [ ]:
```
# hence we have plotted first parameter b=[80,130,210] on graph for data visualisation, now consider second parameter,
```

In [6]:
```
c=[1,2,3]
d=[85,145,205]
plt.bar(c,d)
plt.show()
```



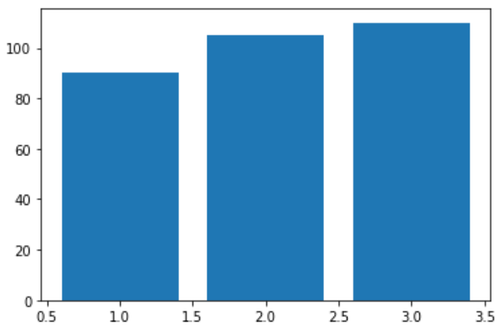

In [ ]:
```
# hence we have plotted second parameter d=[85,145,205] on graph for data visualisation to machine now consider third parameter
```

In [7]:
```
e=[1,2,3]
f=[115,155,320]
plt.bar(e,f)
plt.show()
```



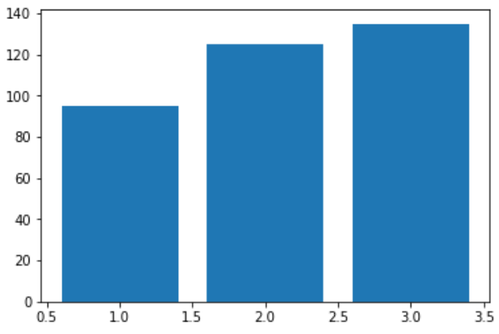

In [ ]:
```
# hence we have plotted 3rd parameter f=[115,155,320] on graph for data visualisation, now consider 4th parameter
```

In [8]:
```
g=[1,2,3]
h=[90,105,110]
plt.bar(g,h)
plt.show()
```




In [ ]:
```
# hence we have calculated forth parameter, now consider 5th parameter
```

In [9]:
```
i=[1,2,3]
j=[95,125,135]
plt.bar(i,j)
plt.show()
```




In [ ]:
```
# hence we calculated 5 parameters, now apply machine prediction formula - data matrix * parameters, (before arrange all parameters on matrix)
```

In [10]:
```
arr2=np.array(([80,130,210],[85,145,205],[115,155,320],[90,105,110],[95,125,135]))
```

In [11]:

Out[11]:

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

```
arr2
```

```
array([[ 80, 130, 210],
       [ 85, 145, 205],
       [111, 155, 320],
       [ 90, 105, 110],
       [ 95, 125, 135]])
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Let us consider engine firing order be = 1342, before plug in the equation to machine first arrange data Matrix and parameters.
```

In [1]:
```
import numpy as np
```

In [2]:
```
arr1=np.array(([1,1],[1,3],[1,4],[1,2]))
```

In [3]:
```
arr1
```

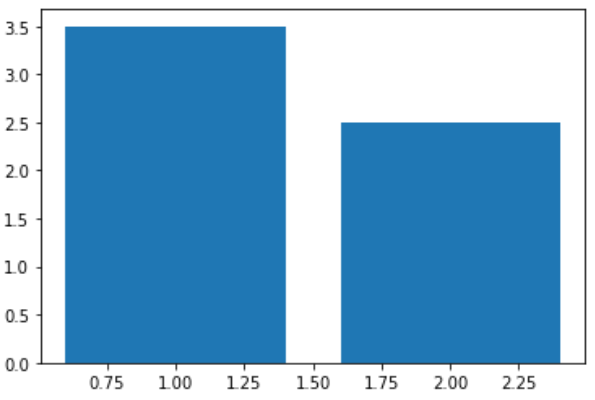
Out[3]:
```
array([[1, 1],
       [1, 3],
       [1, 4],
       [1, 2]])
```

In [ ]:
```
# as above we have arranged data matrix, now calculate atleast 3 parameters for machine learning, h=[h1+h2+h3. ..... nth]
```

In [ ]:
```
# first calculate h1, plot the graph
```

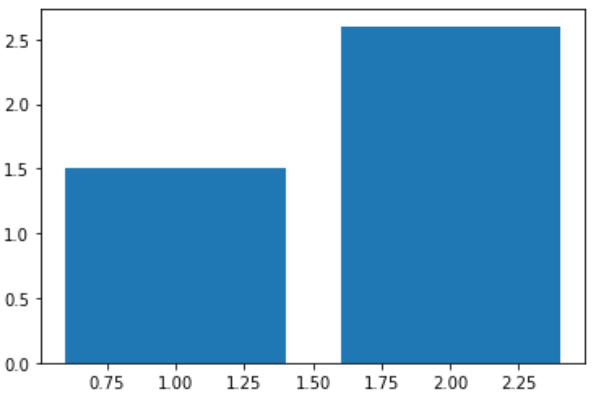
In [4]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
a=[1,2]
b=[3.5,2.5]
plt.bar(a,b)
plt.show()
```



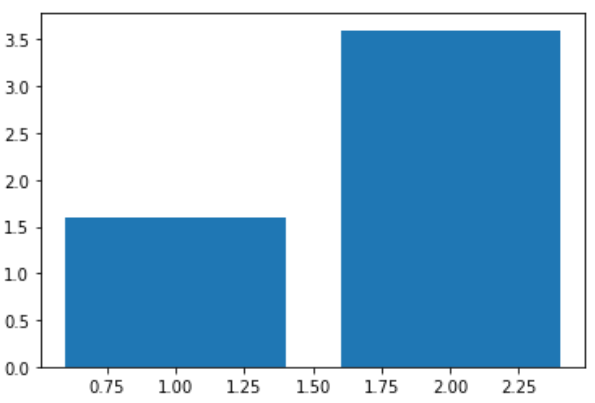

In [ ]:
```
# hence, we got firtst parameters (b=3.5,2.5) plotted on the graph for machine data visualisation, now plot second parameter.
```

In [9]:
```
c=[1,2]
d=[1.5,2.6]
plt.bar(c,d)
plt.show()
```




In [ ]:
```
# hence we have plotted second parameter d=1.2,2.6 on graph for machine learning, now consider third parameter
```

In [10]:
```
e=[1,2]
f=[1.6,3.6]
plt.bar(e,f)
plt.show()
```




In [ ]:
```
# hence we have got third parameter f=[1.6,3.6] plotted on graph for data visualisation, now arrange all the 3 parameters on matrix
```

In [11]:
```
arr2=np.array(([3.5,2.5],[1.6,2.6],[1.6,3.6]))
```

In [12]:
```
arr2
```

Out[12]:
```
array([[3.5, 2.5],
       [1.6, 2.6],
       [1.6, 3.6]])
```

In [ ]:
```
# now apply machine prediction formula = data Matrix * all three hypothesis(paramters) and the equation would be plug in to machine
```

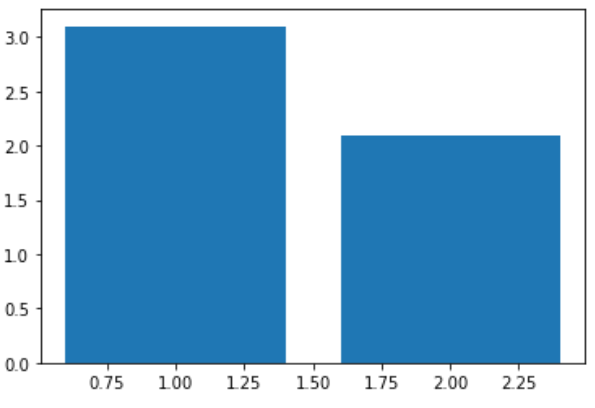
In [13]:
```
arr1*arr2
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [13], in <cell line: 1>()
----> 1 arr1*arr2

ValueError: operands could not be broadcast together with shapes (4,2) (3,2)
```

In [ ]:
```
# dimenison 4,2 & 3,2 hence we need to correct equal dimesions matrix so now consider 4th parameter,
```

In [14]:
```
g=[1,2]
h=[3.1,2.1]
plt.bar(g,h)
plt.show()
```




In [ ]:
```
# hence we got forth parameter g=3.1,2.1, now arrange matrix once again
```

In [15]:
```
arr2=np.array(([3.5,2.6],[1.6,2.6],[1.6,3.6],[3.1,2.1]))
```

In [16]:
```
arr2
```

Out[16]:
```
array([[3.5, 2.6],
       [1.6, 2.6],
       [1.6, 3.6],
       [3.1, 2.1]])
```

In [ ]:
```
# now finally apply machine prediction formula=data Matrix * paramters
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [17]:
```
arr1*arr2
```

Out[17]:
```
array([[ 3.5,  2.6],
       [ 1.6,  7.8],
       [ 1.6, 14.4],
       [ 3.1,  4.2]])
```

In [ ]:

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [ ]:
```
# Let us consider rate of change in position of flywheel with respect to seconds of time (t) be = pow(x,n) (as the rate of change in position is nth  times)
```

In [1]:
```
import sympy as smp
from sympy import *
```

In [5]:
```
x,n = smp.symbols('x n')
```

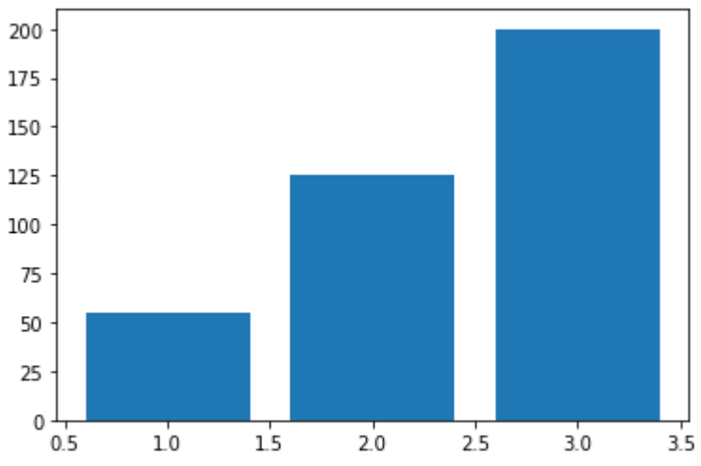
In [9]:
```
smp.diff(pow(x,3)),x
```

Out[9]:
```
(3*x**2, x)
```

In [ ]:
```
# Now lets consider atleast 3 hypothesis for machine learning, h=[h1+h2+h3.....nth], now plot the graph,
```

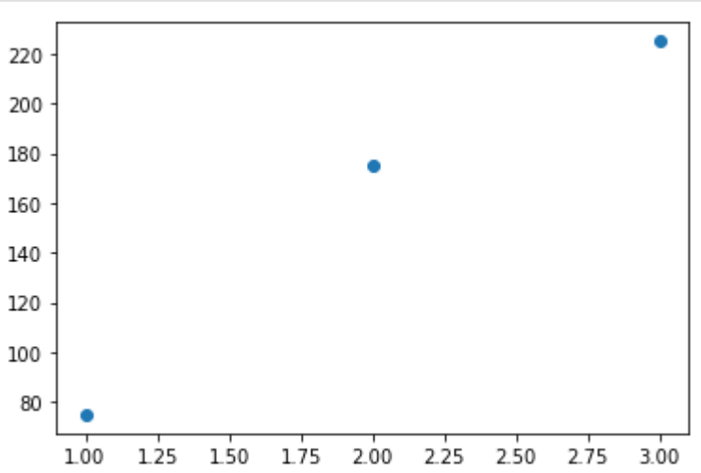
In [10]:
```
import matplotlib.pyplot as plt
```

In [11]:
```
a=[1,2,3]
b=[55,125,200]
plt.bar(a,b)
plt.show()
```



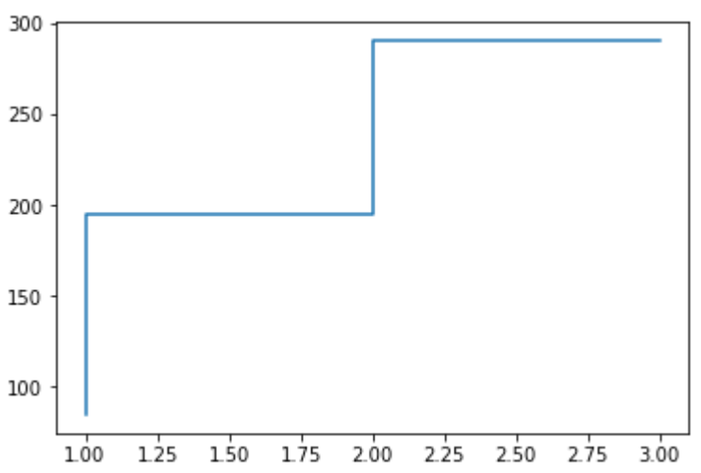

In [ ]:
```
# hence we have successfully plotted first parameter on machine for data visualisation, now consider second parameter,
```

In [12]:
```
c=[1,2,3]
d=[75,175,225]
plt.scatter(c,d)
plt.show()
```




In [ ]:
```
# hence we have successfully plotted second parameter on machine for data visualisation, now let us consider third parameter.
```

In [13]:
```
e=[1,2,3]
f=[85,195,290]
plt.step(e,f)
plt.show()
```




In [ ]:
```
# hence we have successfully plotted third parameter on machine for data visualisation, now Dendrite funciton= Data Matrix * Parametes
```

In [ ]:
```
# arrange all set of parameters on Matrix
```

In [14]:
```
import numpy as np
```

In [15]:
```
arr1=np.array(([55,125,200],[75,175,225],[85,195,290]))
```

In [16]:
```
arr1
```

Out[16]:
```
array([[ 55, 125, 200],
       [ 75, 175, 225],
       [ 85, 195, 290]])
```

In [ ]:
```
# now data matrix * parameters
```

In [ ]:
```

```

# " Car ECU Nueral Scheme"

Prem Kumar
Charter Automobile Engineer Bihar"

In [17]:
```
smp.diff(pow(x,3)),x*arr1
```

Out[17]:
```
(3*x**2,
 array([[55*x, 125*x, 200*x],
        [75*x, 175*x, 225*x],
        [85*x, 195*x, 290*x]], dtype=object))
```

In [ ]:
```

```