In [2]:
```
import matplotlib as m
```

In [3]:
```
import numpy as np

#Seasons
Seasons = ["2010","2011","2012","2013","2014","2015","2016","2017","2018","2019"
Sdict = {"2010":0,"2011":1,"2012":2,"2013":3,"2014":4,"2015":5,"2016":6,"2017":7

#Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Samson"

#Salaries
Sachin_Salary = [15946875,17718750,19490625,21262500,23034375,24806250,25244493,
Rahul_Salary = [12000000,12744189,13488377,14232567,14976754,16324500,18038573,1
Smith_Salary = [4621800,5828090,13041250,14410581,15779912,14500000,16022500,175
Sami_Salary = [3713640,4694041,13041250,14410581,15779912,17149243,18518574,1945
Pollard_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,18091770,19
Morris_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,16022500,17
Samson_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,1777
Dhoni_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,1
Kohli_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875
Sky_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182
#Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Polla

#Games
Sachin_G = [80,77,82,82,73,82,58,78,6,35]
Rahul_G = [82,57,82,79,76,72,60,72,79,80]
Smith_G = [79,78,75,81,76,79,62,76,77,69]
Sami_G = [80,65,77,66,69,77,55,67,77,40]
Pollard_G = [82,82,82,79,82,78,54,76,71,41]
Morris_G = [70,69,67,77,70,77,57,74,79,44]
Samson_G = [78,64,80,78,45,80,60,70,62,82]
Dhoni_G = [35,35,80,74,82,78,66,81,81,27]
Kohli_G = [40,40,40,81,78,81,39,0,10,51]
Sky_G = [75,51,51,79,77,76,49,69,54,62]
#Matrix
Games = np.array([Sachin_G, Rahul_G, Smith_G, Sami_G, Pollard_G, Morris_G, Samsc

#Points
Sachin_PTS = [2832,2430,2323,2201,1970,2078,1616,2133,83,782]
Rahul_PTS = [1653,1426,1779,1688,1619,1312,1129,1170,1245,1154]
Smith_PTS = [2478,2132,2250,2304,2258,2111,1683,2036,2089,1743]
Sami_PTS = [2122,1881,1978,1504,1943,1970,1245,1920,2112,966]
Pollard_PTS = [1292,1443,1695,1624,1503,1784,1113,1296,1297,646]
Morris_PTS = [1572,1561,1496,1746,1678,1438,1025,1232,1281,928]
Samson_PTS = [1258,1104,1684,1781,841,1268,1189,1186,1185,1564]
Dhoni_PTS = [903,903,1624,1871,2472,2161,1850,2280,2593,686]
Kohli_PTS = [597,597,597,1361,1619,2026,852,0,159,904]
Sky_PTS = [2040,1397,1254,2386,2045,1941,1082,1463,1028,1331]
#Matrix
Points = np.array([Sachin_PTS, Rahul_PTS, Smith_PTS, Sami_PTS, Pollard_PTS, Morr
```

In [4]:
```
Seasons
```

Out[4]:
```
['2010',
 '2011',
 '2012',
 '2013',
 '2014',
 '2015',
 '2016',
 '2017',
 '2018',
 '2019']
```

In [5]:
```
Salary
```

Out[5]:
```
array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000],
       [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
        18038573, 19752645, 21466718, 23180790],
       [ 4621800,  5828090, 13041250, 14410581, 15779912, 14500000,
        16022500, 17545000, 19067500, 20644400],
       [ 3713640,  4694041, 13041250, 14410581, 15779912, 17149243,
        18518574, 19450000, 22407474, 22458000],
       [ 4493160,  4806720,  6061274, 13758000, 15202590, 16647180,
        18091770, 19536360, 20513178, 21436271],
       [ 3348000,  4235220, 12455000, 14410581, 15779912, 14500000,
        16022500, 17545000, 19067500, 20644400],
       [ 3144240,  3380160,  3615960,  4574189, 13520500, 14940153,
        16359805, 17779458, 18668431, 20068563],
       [       0,        0,  4171200,  4484040,  4796880,  6053663,
        15506632, 16669630, 17832627, 18995624],
       [       0,        0,        0,  4822800,  5184480,  5546160,
         6993708, 16402500, 17632688, 18862875],
       [ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
        15691000, 17182000, 18673000, 15000000]])
```

In [6]:
```
Games
```

Out[6]:
```
array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],
       [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
       [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
       [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
       [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],
       [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],
       [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],
       [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
       [40, 40, 40, 81, 78, 81, 39,  0, 10, 51],
       [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]])
```

In [7]:
```
Points
```

Out[7]:
```
array([[2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,   83,  782],
       [1653, 1426, 1779, 1688, 1619, 1312, 1129, 1170, 1245, 1154],
       [2478, 2132, 2250, 2304, 2258, 2111, 1683, 2036, 2089, 1743],
       [2122, 1881, 1978, 1504, 1943, 1970, 1245, 1920, 2112,  966],
       [1292, 1443, 1695, 1624, 1503, 1784, 1113, 1296, 1297,  646],
       [1572, 1561, 1496, 1746, 1678, 1438, 1025, 1232, 1281,  928],
       [1258, 1104, 1684, 1781,  841, 1268, 1189, 1186, 1185, 1564],
       [ 903,  903, 1624, 1871, 2472, 2161, 1850, 2280, 2593,  686],
       [ 597,  597,  597, 1361, 1619, 2026,  852,    0,  159,  904],
       [2040, 1397, 1254, 2386, 2045, 1941, 1082, 1463, 1028, 1331]])
```

In [8]:
```
Pdict
```

Out[8]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [9]:
```
Salary/Games
```

```
C:\Users\AR ANSARI\AppData\Local\Temp\ipykernel_7192\3709746658.py:1: RuntimeWarning:
divide by zero encountered in divide
  Salary/Games
```

Out[9]:
```
array([[ 199335.9375    ,  230113.63636364,  237690.54878049,
         259298.7804878 ,  315539.38356164,  302515.24390244,
         435249.87931034,  357040.37179487, 5075634.16666667,
         671428.57142857],
       [ 146341.46341463,  223582.26315789,  164492.40243902,
         180159.07594937,  197062.55263158,  226729.16666667,
         300642.88333333,  274342.29166667,  271730.60759494,
         289759.875     ],
       [  58503.79746835,   74719.1025641 ,  173883.33333333,
         177908.40740741,  207630.42105263,  183544.30379747,
         258427.41935484,  230855.26315789,  247629.87012987,
         299194.20289855],
       [  46420.5       ,   72216.01538462,  169366.88311688,
         218342.13636364,  228694.37681159,  222717.44155844,
         336701.34545455,  290298.50746269,  291006.15584416,
         561450.        ],
       [  54794.63414634,   58618.53658537,   73917.97560976,
         174151.89873418,  185397.43902439,  213425.38461538,
         335032.77777778,  257057.36842105,  288918.        ,
         522835.87804878],
       [  47828.57142857,   61380.        ,  185895.52238806,
         187150.4025974 ,  225427.31428571,  188311.68831169,
         281096.49122807,  237094.59459459,  241360.75949367,
         469190.90909091],
       [  40310.76923077,   52815.        ,   45199.5       ,
          58643.44871795,  300455.55555556,  186751.9125    ,
         272663.41666667,  253992.25714286,  301103.72580645,
         244738.57317073],
       [      0.        ,       0.        ,   52140.        ,
          60595.13513514,   58498.53658537,   77611.06410256,
         234948.96969697,  205797.90123457,  220155.88888889,
         703541.62962963],
       [      0.        ,       0.        ,       0.        ,
          59540.74074074,   66467.69230769,   68471.11111111,
         179325.84615385,              inf, 1763268.8       ,
         369860.29411765],
       [  40425.6       ,   75322.41176471,  255710.78431373,
         182412.41772152,  204933.92207792,  186842.10526316,
         320224.48979592,  249014.49275362,  345796.2962963 ,
         241935.48387097]])
```

In [10]:
```
Salary//Games
```

```
C:\Users\AR ANSARI\AppData\Local\Temp\ipykernel_7192\1634212085.py:1: RuntimeWarning:
divide by zero encountered in floor_divide
  Salary//Games
```

Out[10]:
```
array([[ 199335,  230113,  237690,  259298,  315539,  302515,  435249,
          357040, 5075634,  671428],
       [ 146341,  223582,  164492,  180159,  197062,  226729,  300642,
          274342,  271730,  289759],
       [  58503,   74719,  173883,  177908,  207630,  183544,  258427,
          230855,  247629,  299194],
       [  46420,   72216,  169366,  218342,  228694,  222717,  336701,
          290298,  291006,  561450],
       [  54794,   58618,   73917,  174151,  185397,  213425,  335032,
          257057,  288918,  522835],
       [  47828,   61380,  185895,  187150,  225427,  188311,  281096,
          237094,  241360,  469190],
       [  40310,   52815,   45199,   58643,  300455,  186751,  272663,
          253992,  301103,  244738],
       [      0,       0,   52140,   60595,   58498,   77611,  234948,
          205797,  220155,  703541],
       [      0,       0,       0,   59540,   66467,   68471,  179325,
                0, 1763268,  369860],
       [  40425,   75322,  255710,  182412,  204933,  186842,  320224,
          249014,  345796,  241935]])
```

In [11]:
```
np.round(Salary//Games)
```

```
C:\Users\AR ANSARI\AppData\Local\Temp\ipykernel_7192\3663165759.py:1: RuntimeWarning:
divide by zero encountered in floor_divide
  np.round(Salary//Games)
```

Out[11]:
```
array([[ 199335,  230113,  237690,  259298,  315539,  302515,  435249,
          357040, 5075634,  671428],
       [ 146341,  223582,  164492,  180159,  197062,  226729,  300642,
          274342,  271730,  289759],
       [  58503,   74719,  173883,  177908,  207630,  183544,  258427,
          230855,  247629,  299194],
       [  46420,   72216,  169366,  218342,  228694,  222717,  336701,
          290298,  291006,  561450],
       [  54794,   58618,   73917,  174151,  185397,  213425,  335032,
          257057,  288918,  522835],
       [  47828,   61380,  185895,  187150,  225427,  188311,  281096,
          237094,  241360,  469190],
       [  40310,   52815,   45199,   58643,  300455,  186751,  272663,
          253992,  301103,  244738],
       [      0,       0,   52140,   60595,   58498,   77611,  234948,
          205797,  220155,  703541],
       [      0,       0,       0,   59540,   66467,   68471,  179325,
                0, 1763268,  369860],
       [  40425,   75322,  255710,  182412,  204933,  186842,  320224,
          249014,  345796,  241935]])
```

In [12]:
```
import warnings
warnings.filterwarnings ("ignore")
```

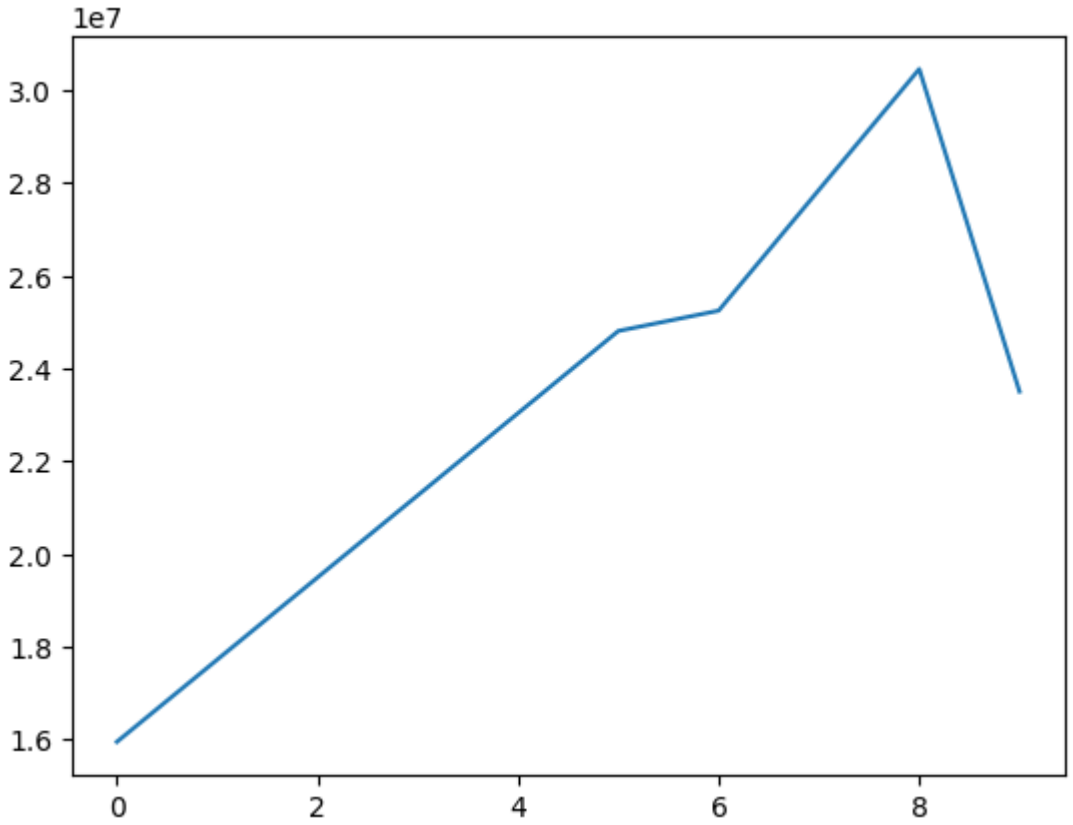
In [13]:
```
import matplotlib.pyplot as plt # for vizuization
```

In [14]:
```
Salary[0]
```

Out[14]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

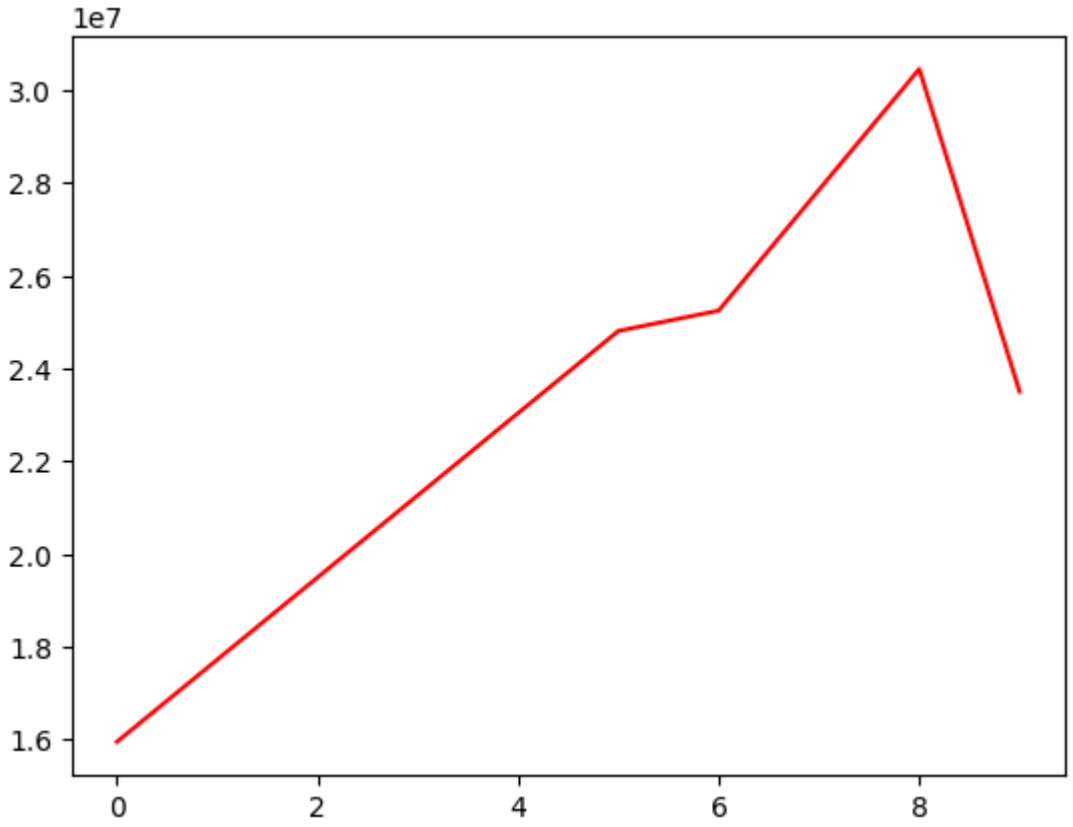
In [15]:
```
plt.plot(Salary[0])
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x19a7ab15a90>]
```



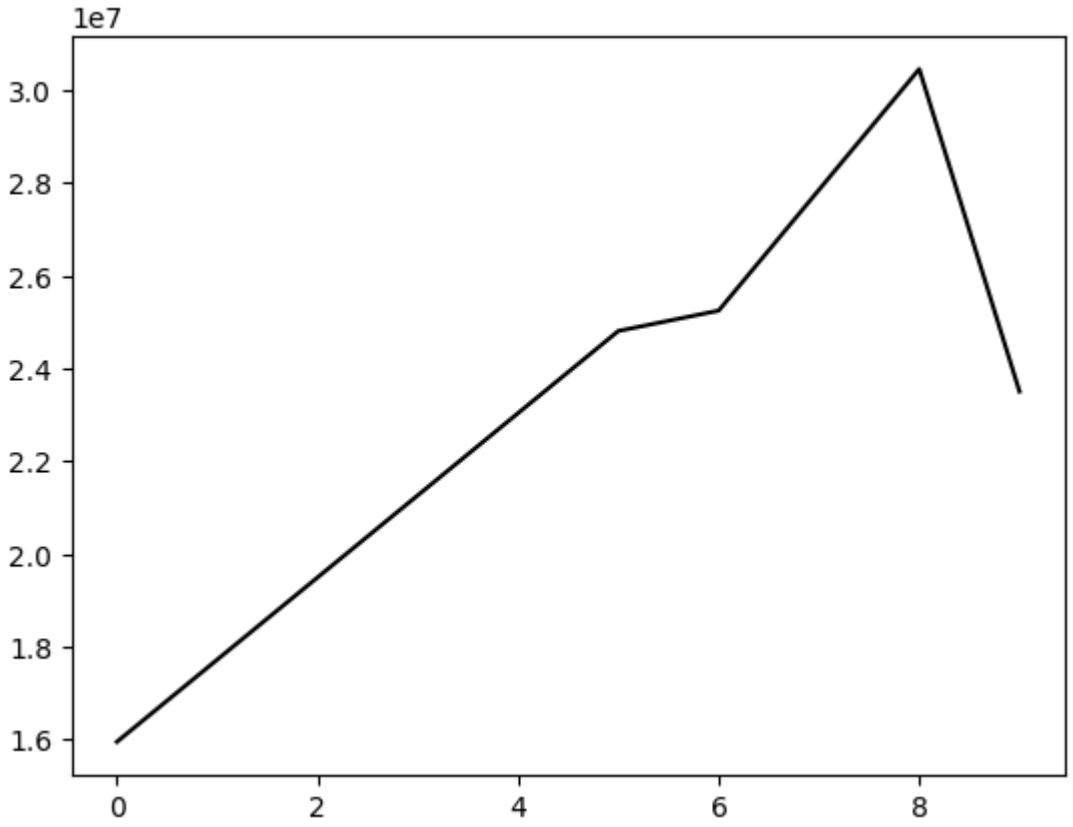
In [16]:
```
plt.plot(Salary[0], color ='r')
```

Out[16]:
```
[<matplotlib.lines.Line2D at 0x19a7cc8a0d0>]
```



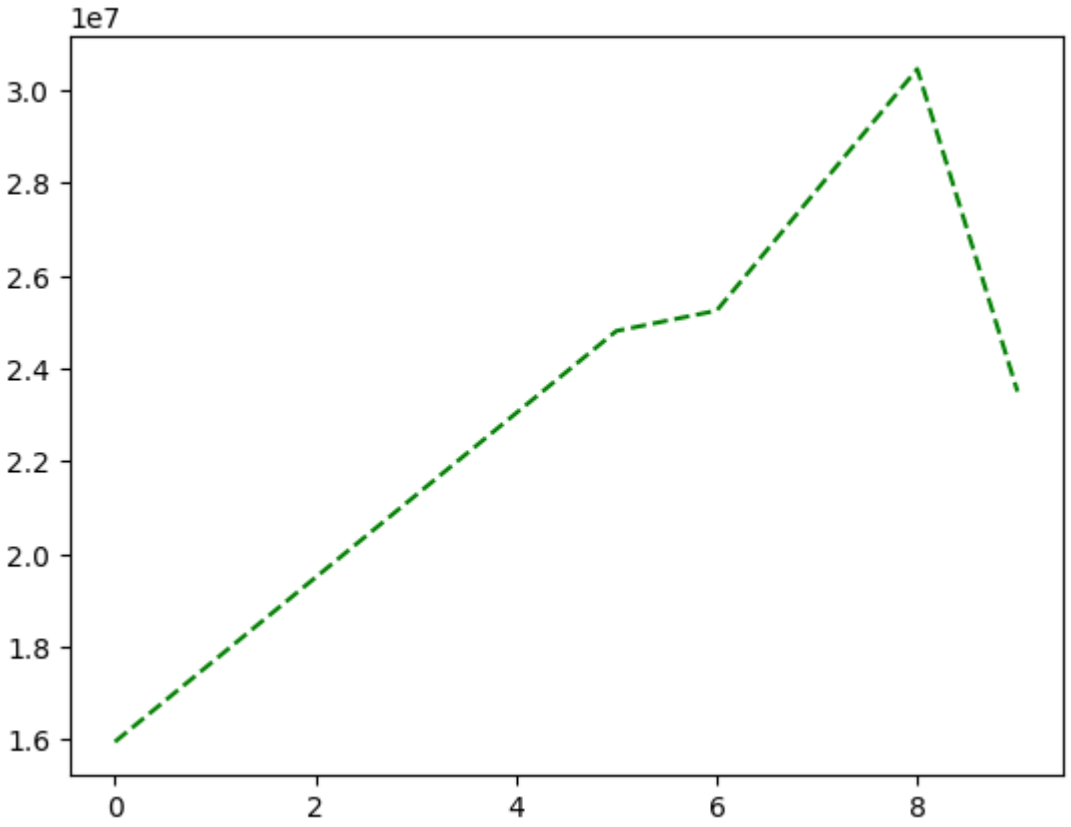
In [17]:
```
plt.plot(Salary[0], color ='k')
```

Out[17]:
```
[<matplotlib.lines.Line2D at 0x19a7cd11090>]
```



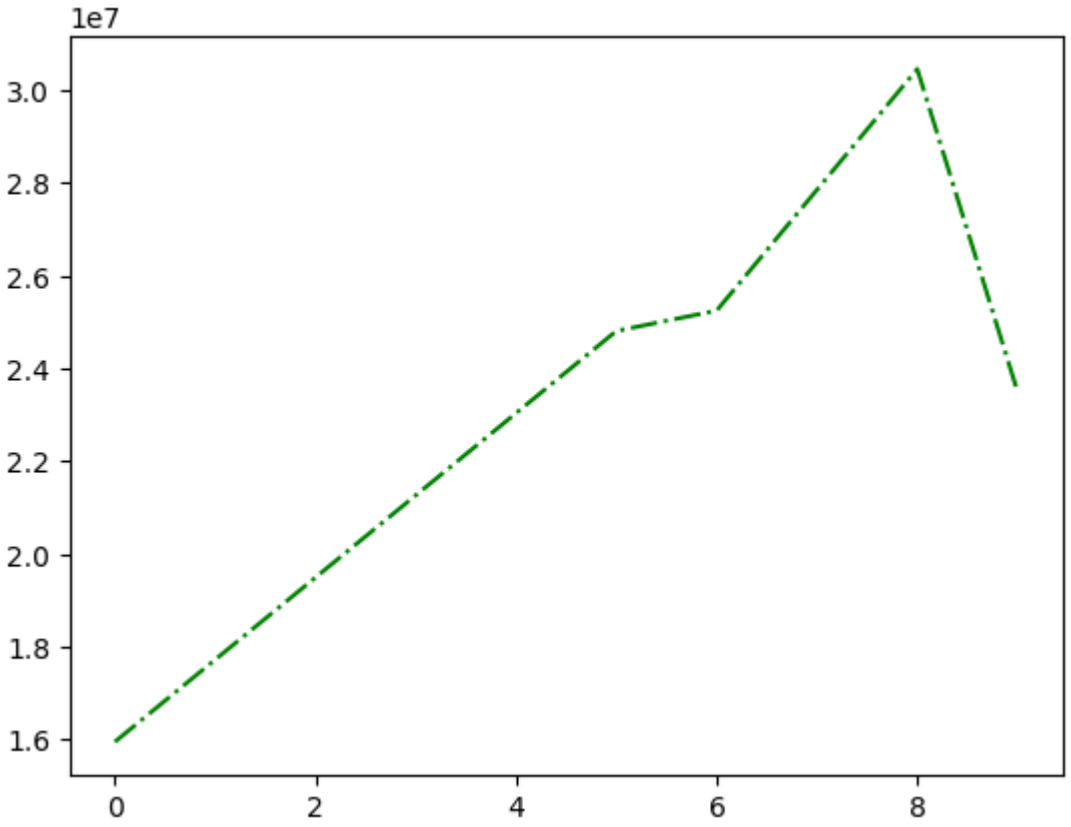
In [18]:
```
plt.plot(Salary[0], color ='g', ls ='--')
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x19a7cd5f610>]
```



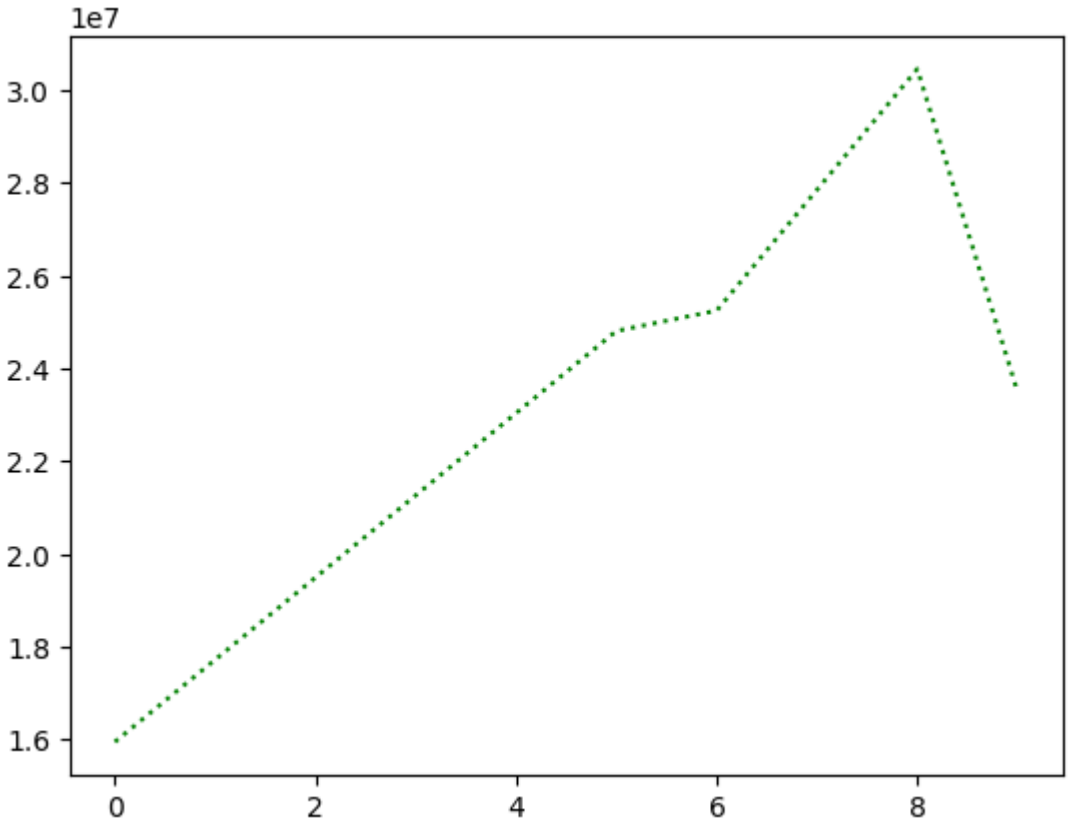
In [19]:
```
plt.plot(Salary[0], color ='g', ls ='-.')
```

Out[19]:
```
[<matplotlib.lines.Line2D at 0x19a7ce09bd0>]
```



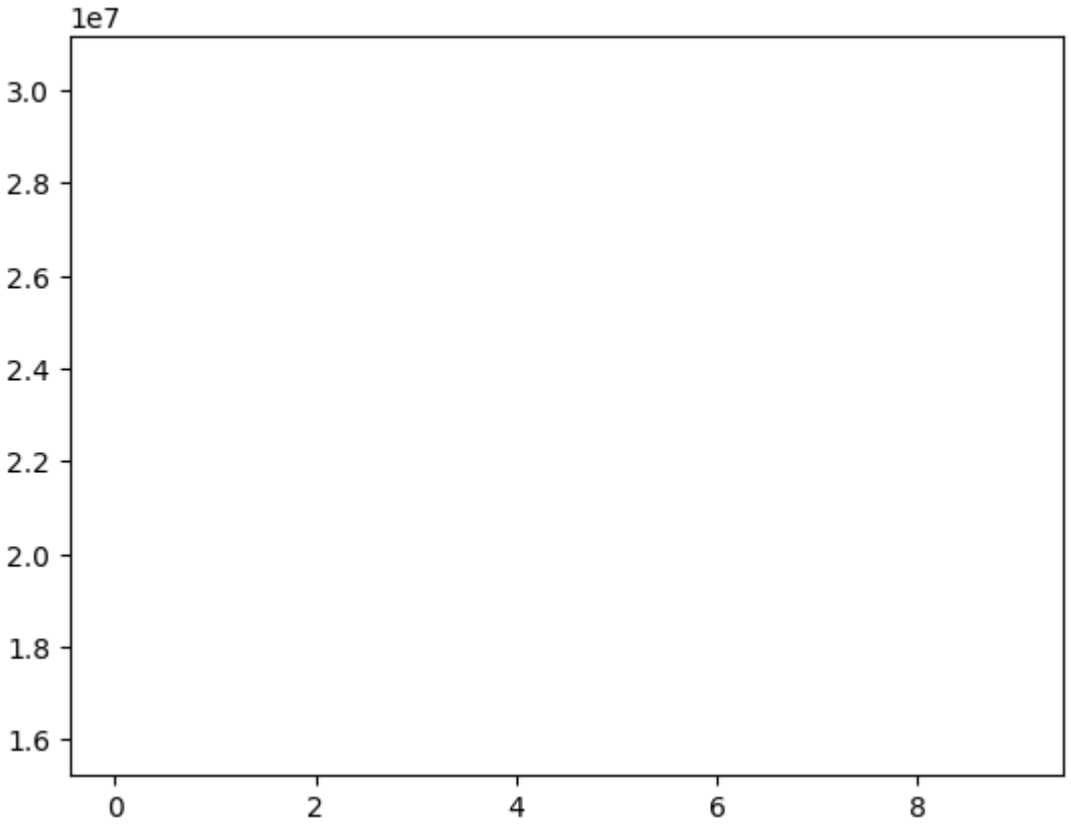
In [20]:
```
plt.plot(Salary[0], color ='g', ls =':')
```

Out[20]:
```
[<matplotlib.lines.Line2D at 0x19a7ce8c190>]
```



In [21]:
```
plt.plot(Salary[0], color ='g', ls ='None')
```

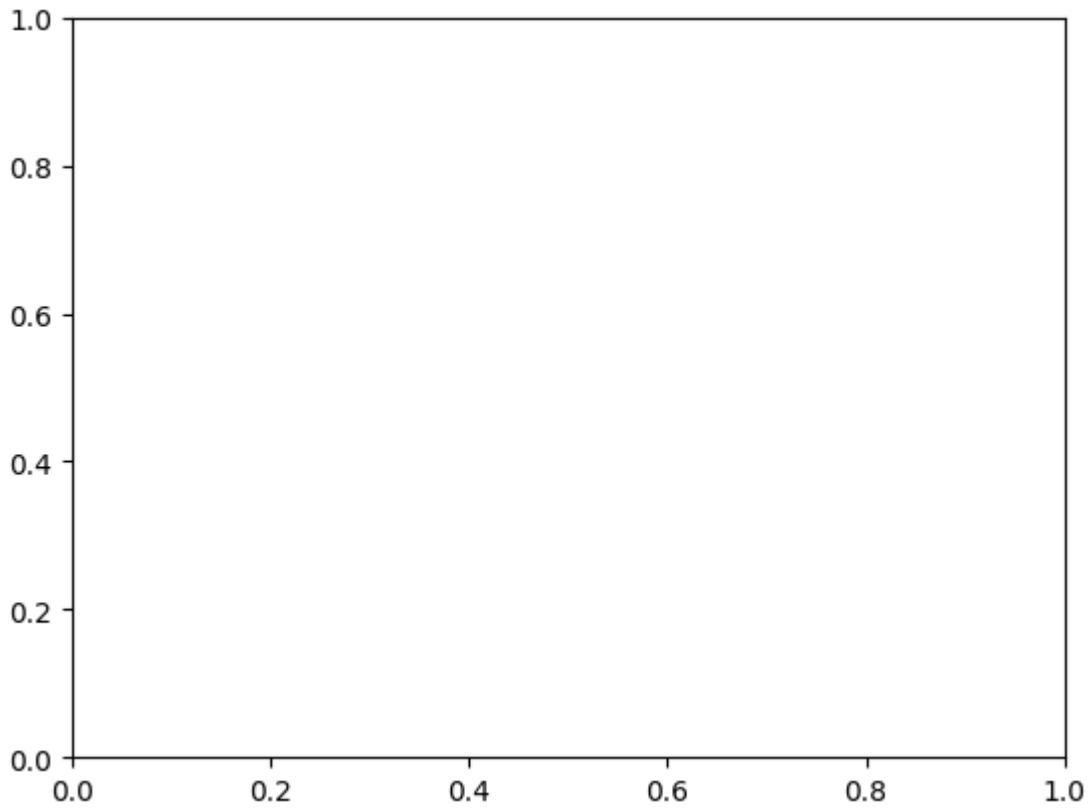
Out[21]:
```
[<matplotlib.lines.Line2D at 0x19a7cee6710>]
```



In [22]:
```
plt.plot(Salary[0], color ='g', ls ='**')
```

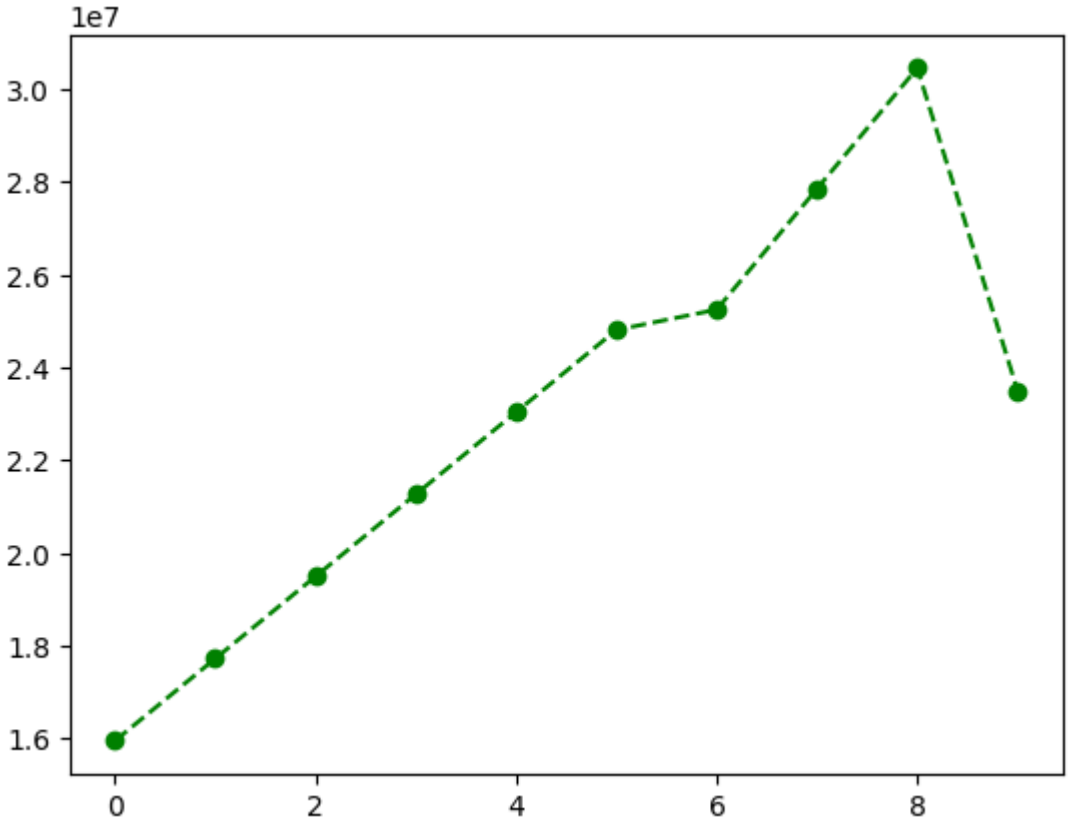
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[22], line 1
----> 1 plt.plot(Salary[0], color =    , ls =     )

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\pyplot.py:3838
lot(scalex, scaley, data, *args, **kwargs)
   3830 @_copy_docstring_and_deprecators(Axes.plot)
   3831 def plot(
   3832     *args: float | ArrayLike | str,
   (...)   3836     **kwargs,
   3837 ) -> list[Line2D]:
-> 3838     return gca().plot(
   3839         *args,
   3840         scalex=scalex,
   3841         scaley=scaley,
   3842         **({      : data} if data is not None else {}),
   3843         **kwargs,
   3844     )

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axes\_axes.py
in Axes.plot(self, scalex, scaley, data, *args, **kwargs)
   1534 """
   1535 Plot y versus x as lines and/or markers.
   1536
   (...)   1774 (``'green'``) or hex strings (``'#008000'``).
   1775 """
   1776 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1777 lines = [*self._get_lines(self, *args, data=data, **kwargs)]
   1778 for line in lines:
   1779     self.add_line(line)

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axes\_base.py
in _process_plot_var_args.__call__(self, axes, data, return_kwargs, *args, **kwargs)
    295     this += args[0],
    296     args = args[1:]
--> 297 yield from self._plot_args(
    298     axes, this, kwargs, ambiguous_fmt_datakey=ambiguous_fmt_datakey,
    299     return_kwargs=return_kwargs
    300 )

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axes\_base.py
in _process_plot_var_args._plot_args(self, axes, tup, kwargs, return_kwargs,
ambiguous_fmt_datakey)
    544     return list(result)
    545 else:
--> 546     return [l[0] for l in result]

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axes\_base.py
in <genexpr>(.0)
    534 else:
    535     raise ValueError(
    536         f"label must be scalar or have the same length as the input "
    537         f"data, but found {len(label)} for {n_datasets} datasets.")
--> 539 result = (make_artist(axes, x[:, j % ncx], y[:, j % ncy], kw,
    540                       {**kwargs,        : label})
    541           for j, label in enumerate(labels))
    543 if return_kwargs:
    544     return list(result)
```

```
File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axes\_base.py
in _process_plot_var_args._make_line(self, axes, x, y, kw, kwargs)
    336 kw = {**kw, **kwargs}  # Don't modify the original kw.
    337 self._setdefaults(self._getdefaults(kw), kw)
--> 338 seg = mlines.Line2D(x, y, **kw)
    339 return seg, kw

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\lines.py:386,
e2D.__init__(self, xdata, ydata, linewidth, linestyle, color, gapcolor, marker, marke
markeredgewidth, markeredgecolor, markerfacecolor, markerfacecoloralt, fillstyle,
antialiased, dash_capstyle, solid_capstyle, dash_joinstyle, solid_joinstyle, pickradi
drawstyle, markevery, **kwargs)
    383 self._dash_pattern = (0, None)  # offset, dash (scaled by linewidth)
    385 self.set_linewidth(linewidth)
--> 386 self.set_linestyle(linestyle)
    387 self.set_drawstyle(drawstyle)
    389 self._color = None

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\lines.py:1192,
ne2D.set_linestyle(self, ls)
   1190 if ls in [' ', '', 'none']:
   1191     ls = 'None'
-> 1192 _api.check_in_list([*self._lineStyles, *ls_mapper_r], ls=ls)
   1193 if ls not in self._lineStyles:
   1194     ls = ls_mapper_r[ls]

File c:\Users\AR ANSARI\vscode\playground\Lib\site-
packages\matplotlib\_api\__init__.py:130, in
check_in_list(values, _print_supported_values, **kwargs)
    128 if _print_supported_values:
    129     msg += f"; supported values are {', '.join(map(repr, values))}"
--> 130 raise ValueError(msg)

ValueError: '**' is not a valid value for ls; supported values are '-', '--', '-.',
'None', ' ', '', 'solid', 'dashed', 'dashdot', 'dotted'
```



In [ ]:
```
plt.plot(Salary[0], color ='g', ls ='--', marker ='o')
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x1d1198a20d0>]
```



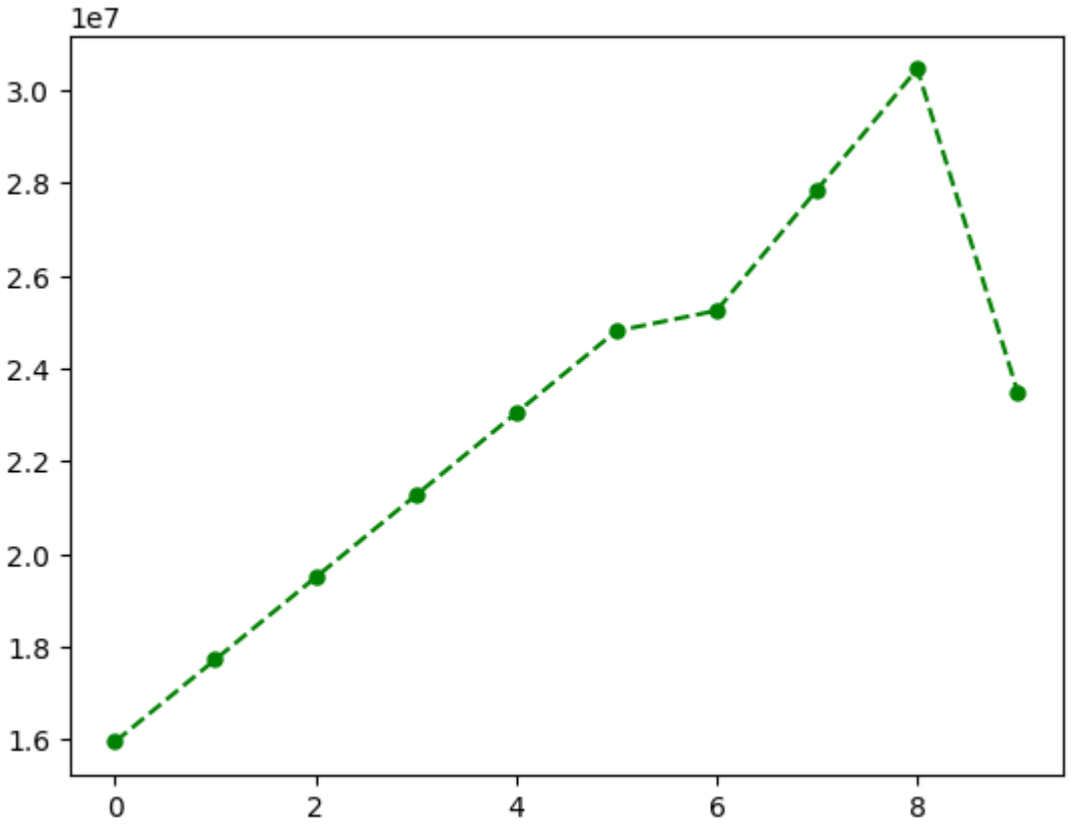
In [ ]:
```
Games[0]
```

Out[ ]:
```
array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

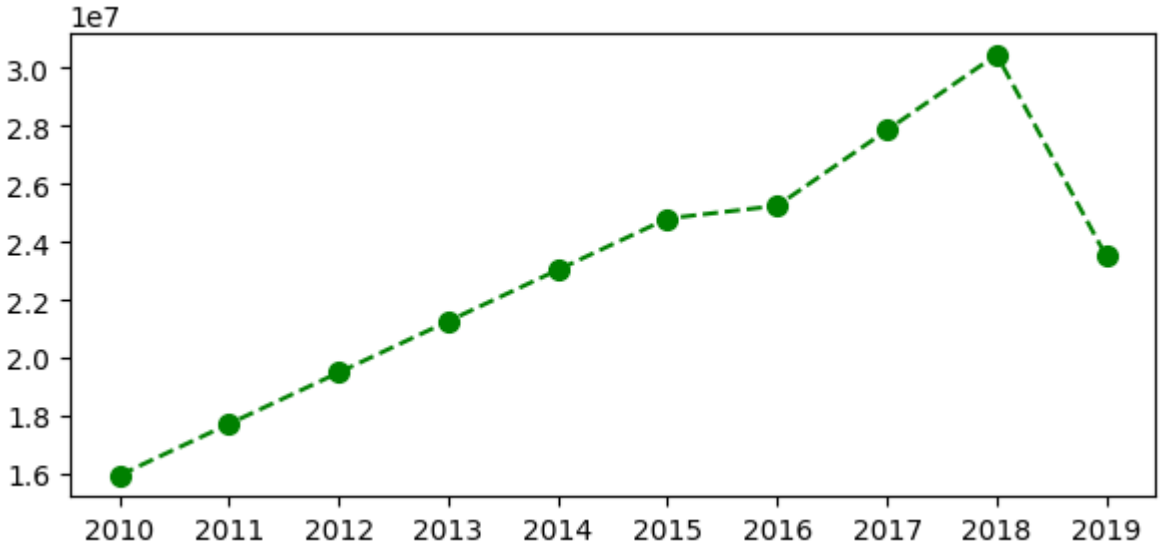
In [ ]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 7,3         # set figure size
```

In [ ]:
```
plt.plot(Salary[0], color ='g', ls ='--', marker ='o',ms = 5) # for size (ms) or
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x1f497b8a350>]
```



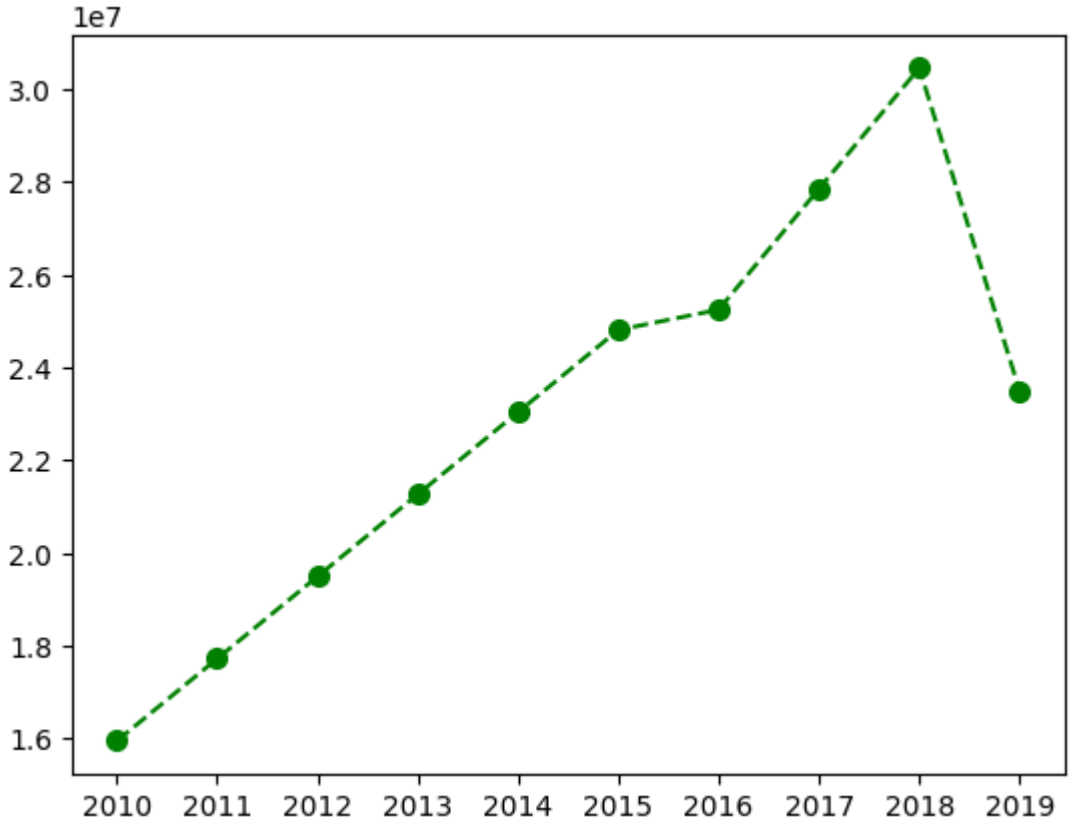
In [ ]:
```
plt.plot(Salary[0], color ='g', ls ='--', marker ='o',ms =7) # for size (ms) on
plt.xticks(list(range(0,10)), Seasons) # set x-ticks to seasons
plt.show() # add year is graph
```



In [ ]:
```
plt.plot(Salary[0], color ='g', ls ='--', marker ='o',ms =7) # for size (ms) on
plt.xticks(list(range(0,10)), Seasons,rotation = 'vertical ') # set x-ticks to s
plt.show()  # add year is graph
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[8], line 2
      1 plt.plot(Salary[0], color ='g', ls ='--', marker ='o',ms =7) # for size (ms)
screen is clear
----> 2 plt.xticks(list(range(0,10)), Seasons,rotation =              ) # set x-ticks t
seasons
      3 plt.show()  # add year is graph

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\pyplot.py:2247
ticks(ticks, labels, minor, **kwargs)
   2245         l._internal_update(kwargs)
   2246 else:
-> 2247     labels_out = ax.set_xticklabels(labels, minor=minor, **kwargs)
   2249 return locs, labels_out

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axes\_base.py:
in _axis_method_wrapper.__set_name__.<locals>.wrapper(self, *args, **kwargs)
     73 def wrapper(self, *args, **kwargs):
---> 74     return get_method(self)(*args, **kwargs)

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\axis.py:2141,
s.set_ticklabels(self, labels, minor, fontdict, **kwargs)
   2139 # deal with label1
   2140 tick.label1.set_text(tick_label)
-> 2141 tick.label1._internal_update(kwargs)
   2142 # deal with label2
   2143 tick.label2.set_text(tick_label)

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\artist.py:1233
rtist._internal_update(self, kwargs)
   1226 def _internal_update(self, kwargs):
   1227     """
   1228     Update artist properties without prenormalizing them, but generating
   1229     errors as if calling `set`.
   1230
   1231     The lack of prenormalization is to maintain backcompatibility.
   1232     """
-> 1233     return self._update_props(
   1234         kwargs,  {cls.__name__}
   1235          {prop_name!r} )

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\artist.py:1209
rtist._update_props(self, props, errfmt)
   1205             if not callable(func):
   1206                 raise AttributeError(
   1207                     errfmt.format(cls=type(self), prop_name=k),
   1208                     name=k)
-> 1209             ret.append(func(v))
   1210 if ret:
   1211     self.pchanged()

File c:\Users\AR ANSARI\vscode\playground\Lib\site-packages\matplotlib\text.py:1244,
t.set_rotation(self, s)
   1242     self._rotation = 90.
   1243 else:
-> 1244     raise ValueError("rotation must be 'vertical', 'horizontal' or "
   1245                      f"a number, not {s}")
   1246 self.stale = True
```

```
ValueError: rotation must be 'vertical', 'horizontal' or a number, not vertical
```



In [24]:
```
Sdict
```

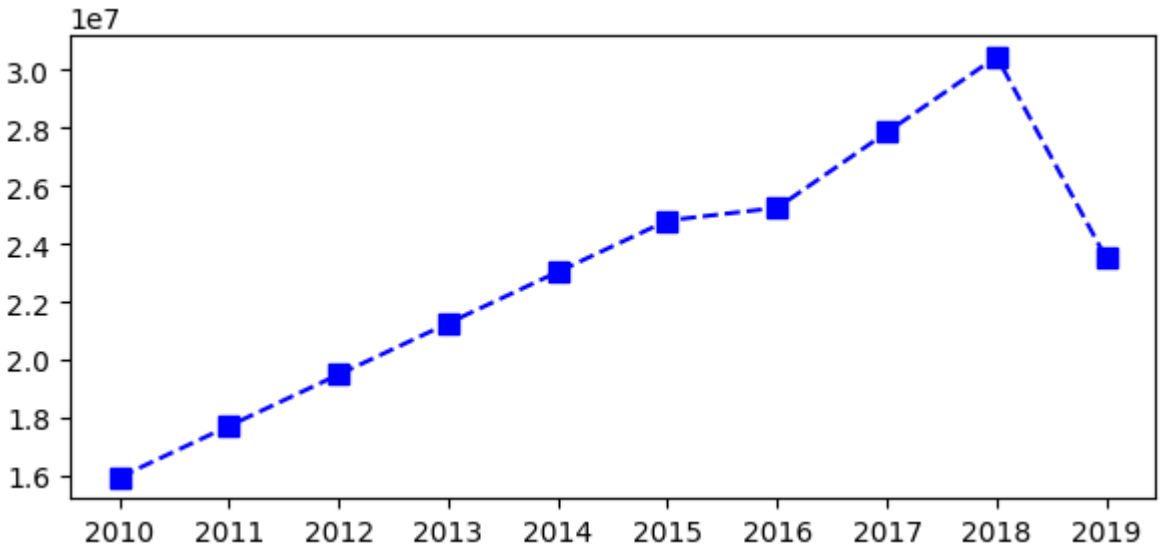
Out[24]:
```
{'2010': 0,
 '2011': 1,
 '2012': 2,
 '2013': 3,
 '2014': 4,
 '2015': 5,
 '2016': 6,
 '2017': 7,
 '2018': 8,
 '2019': 9}
```

In [25]:
```
Pdict
```

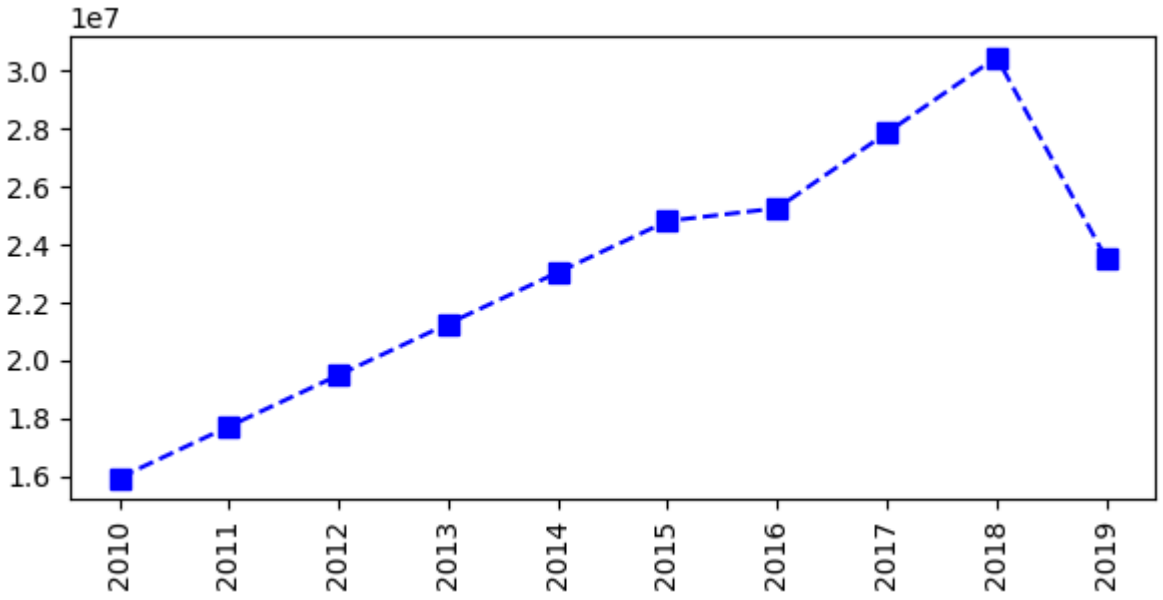
Out[25]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [ ]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 7,3        # set figure size


plt.plot(Salary[0], c='blue', ls ='--',marker ='s',ms='7')
```

```
plt.xticks(list(range(0,10)), Seasons)
plt.show()
```



In [28]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 7,3         # set figure size


plt.plot(Salary[0], c='blue', ls ='--',marker ='s',ms='7')
plt.xticks(list(range(0,10)), Seasons, rotation ='vertical')
plt.show()
```

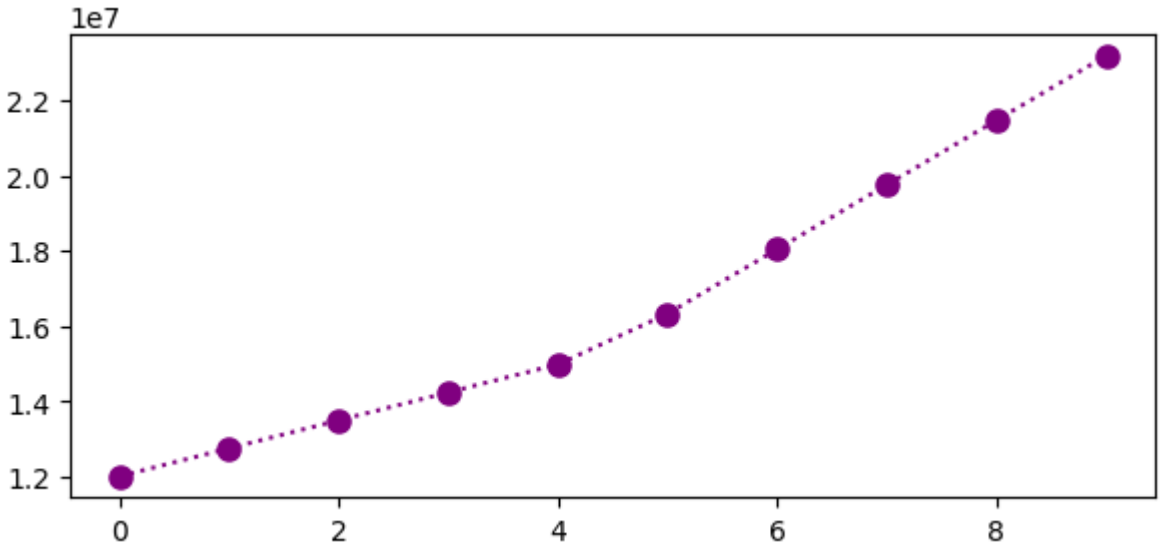


In [29]:
```
Salary[1]
```

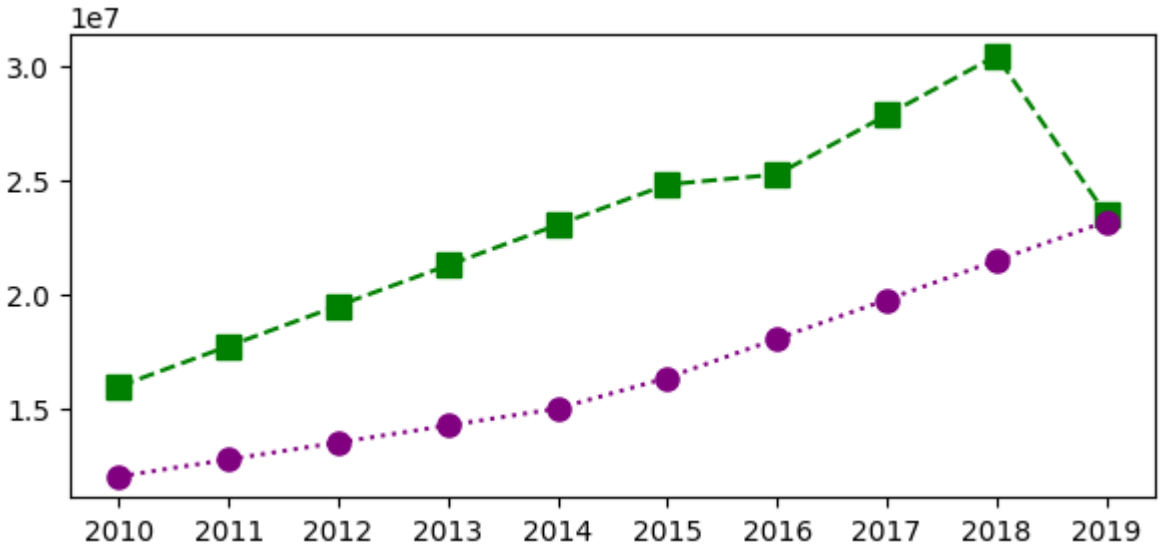
Out[29]:
```
array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
       18038573, 19752645, 21466718, 23180790])
```

In [32]:
```
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
```

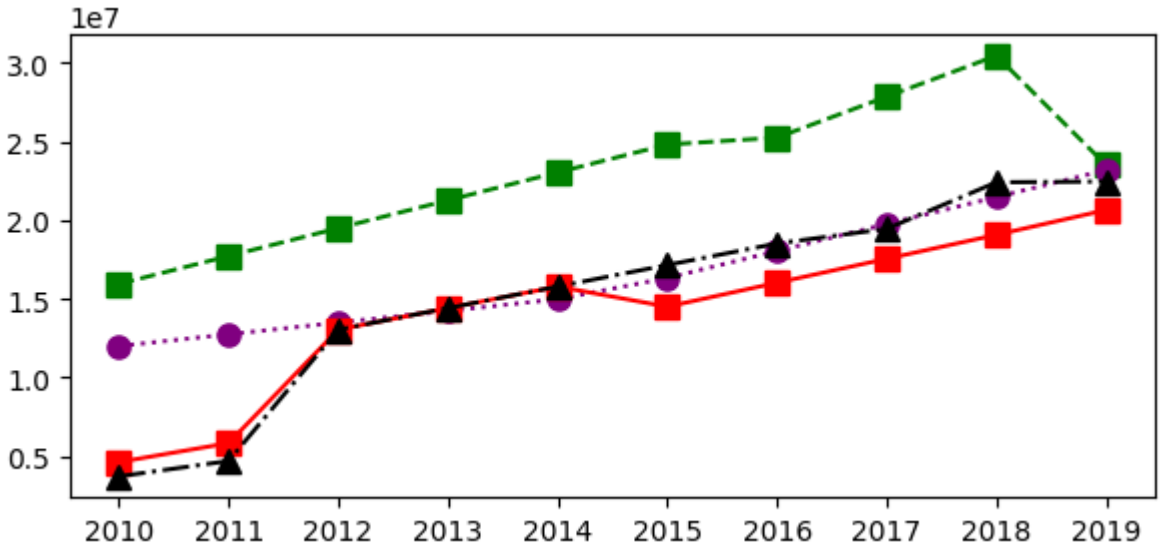
Out[32]:
```
[<matplotlib.lines.Line2D at 0x19a7ff4aad0>]
```



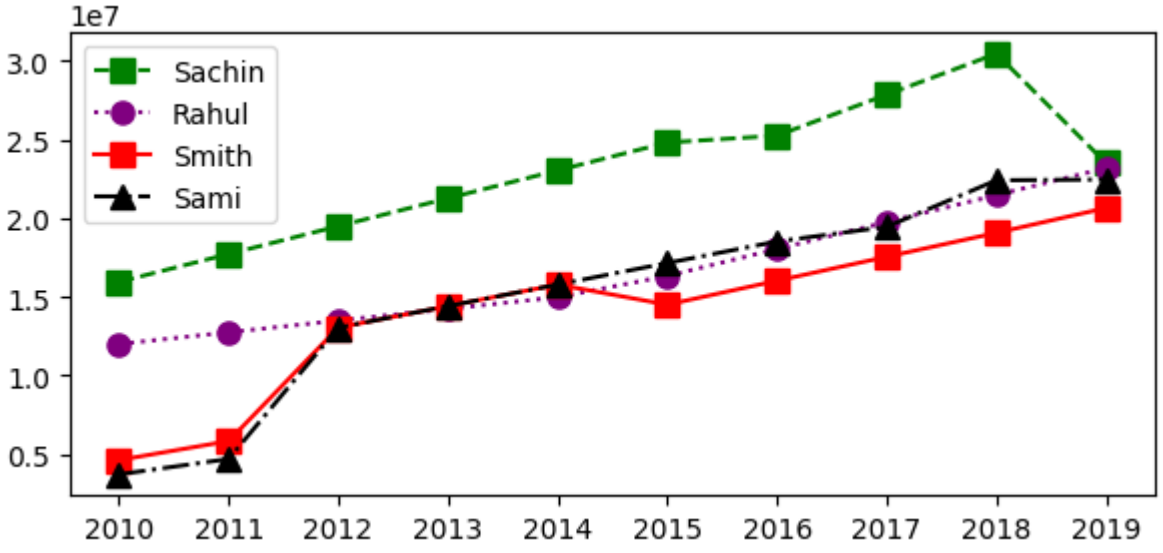
In [ ]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])

plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```



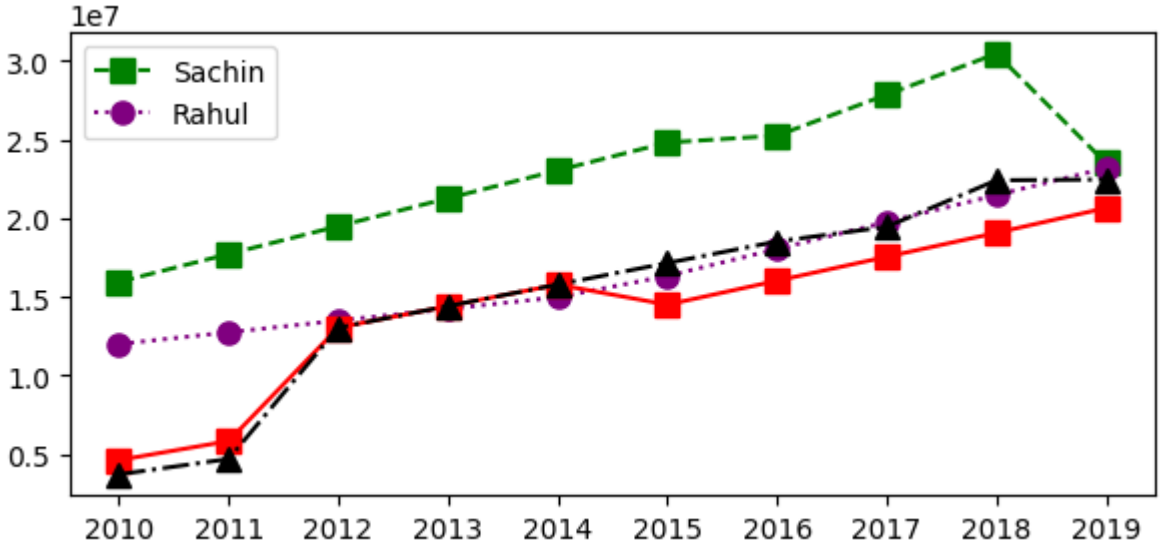
In [39]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```



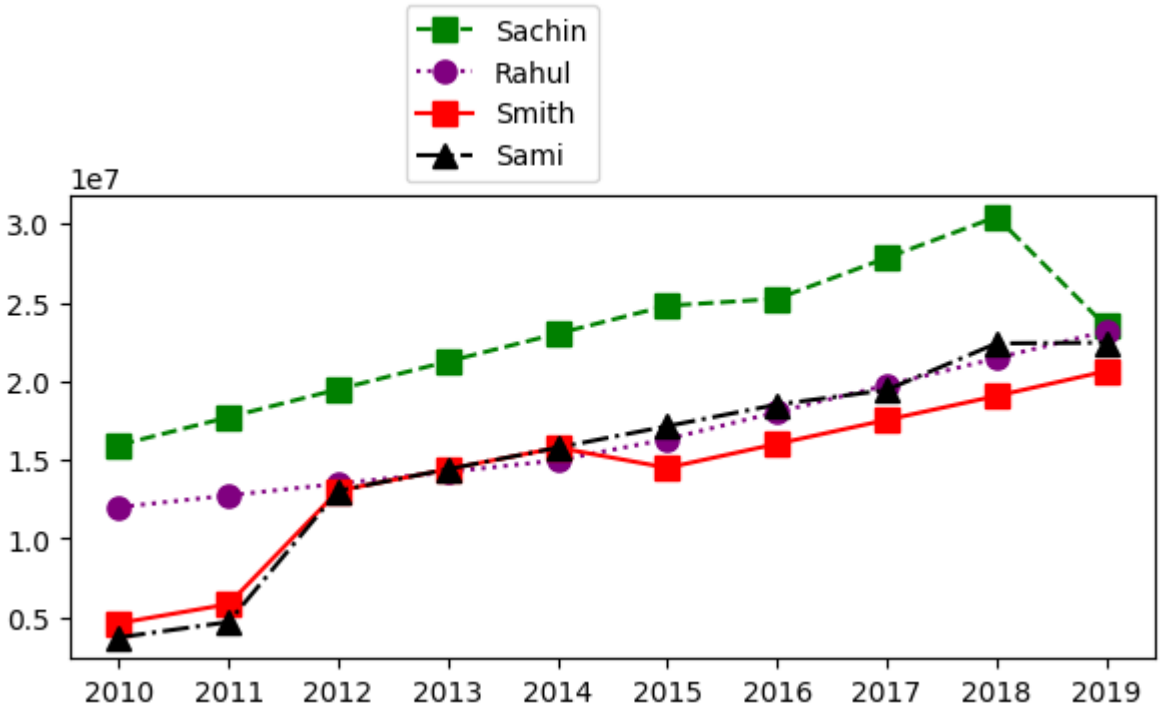
In [ ]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.legend()
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```



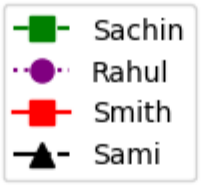
In [41]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8)
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8)

plt.legend()
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```

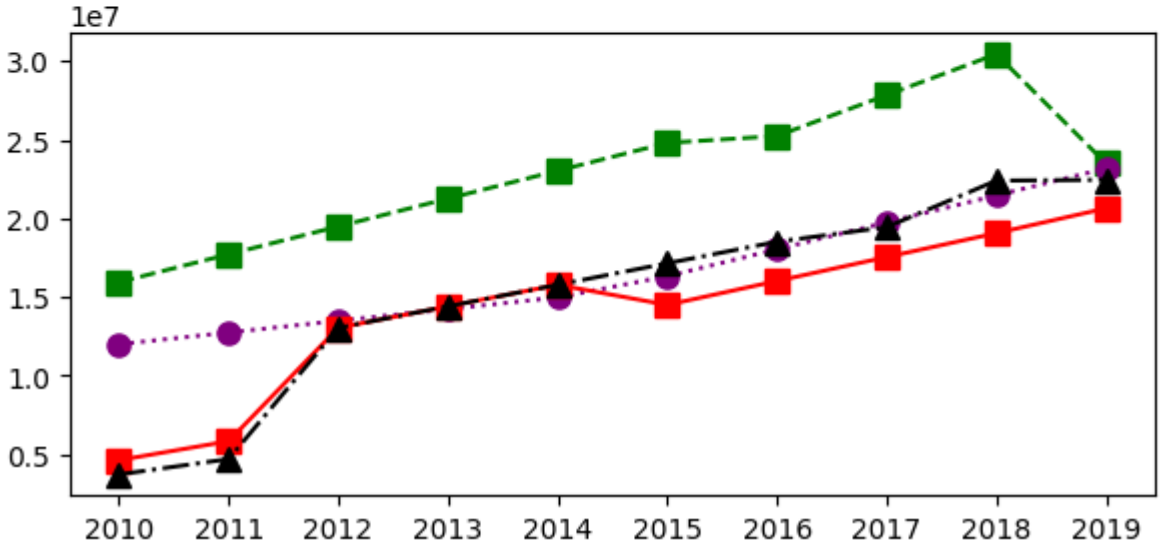


In [42]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1))
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```

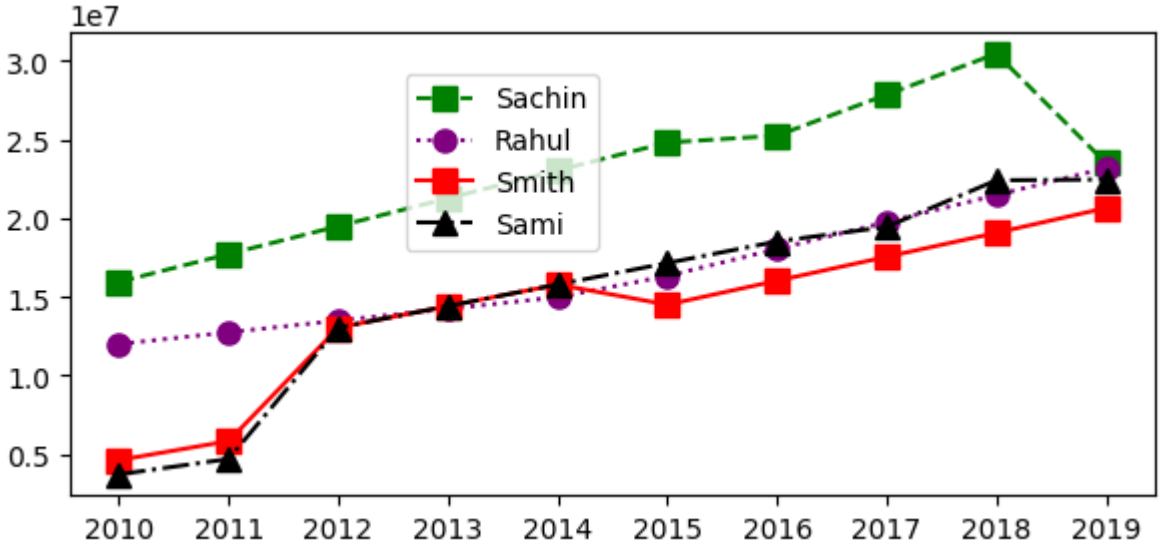


In [43]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,2))
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```

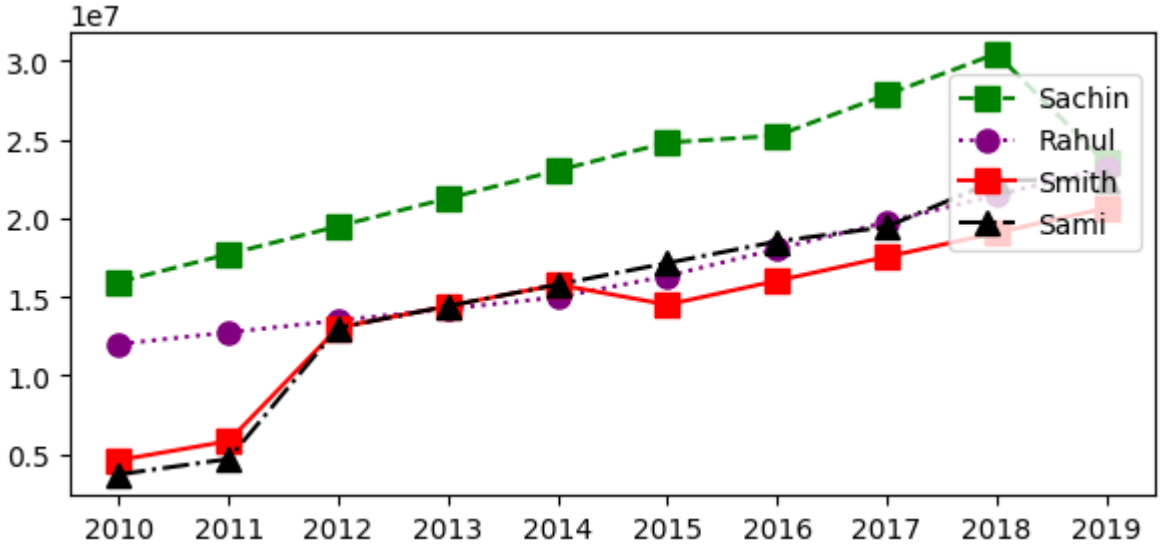




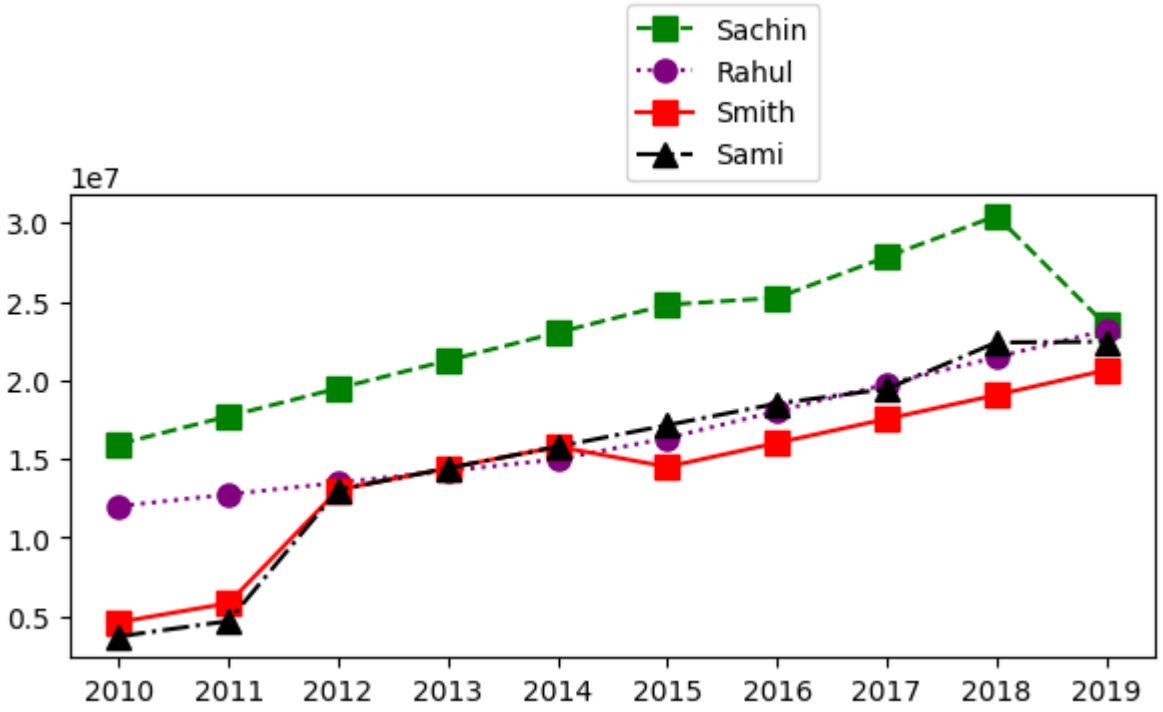
In [44]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,0.5))
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```



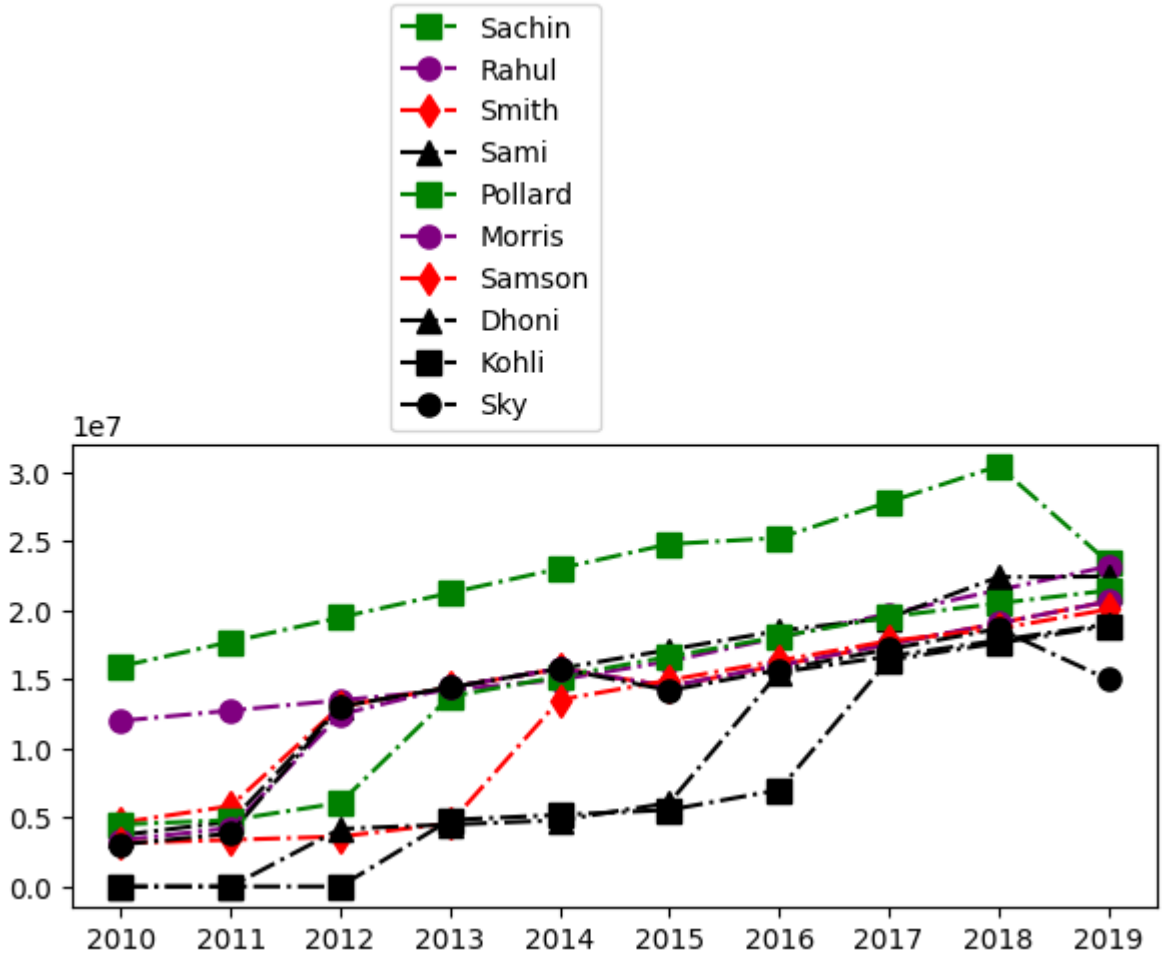
In [45]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.legend(loc = 'lower right',bbox_to_anchor=(1,0.5))
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```



In [47]:
```
plt.plot(Salary[0], c='Green', ls ='--',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls =':',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red', ls ='-',marker ='s', ms =8, label = Players[2])
plt.plot(Salary[3], c='k', ls ='-.',marker ='^', ms =8, label = Players[3])

plt.legend(bbox_to_anchor=(0.5,1))
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```



In [48]:
```
plt.plot(Salary[0], c='Green',  ls ='-.',marker ='s', ms =8, label = Players[0])
plt.plot(Salary[1], c='purple', ls ='-.',marker ='o', ms =8, label = Players[1])
plt.plot(Salary[2], c='red',    ls ='-.',marker ='d',    ms =8,    label = Playe
plt.plot(Salary[3], c='k',      ls ='-.',marker ='^',     ms =8,     label = Pla
plt.plot(Salary[4], c='Green',  ls ='-.',marker ='s', ms =8, label = Players[4])
plt.plot(Salary[5], c='purple', ls ='-.',marker ='o', ms =8, label = Players[5])
plt.plot(Salary[6], c='red',    ls ='-.',marker ='d',    ms =8,    label = Playe
plt.plot(Salary[7], c='k',     ls ='-.',marker ='^',     ms =8,     label = Play
plt.plot(Salary[8], c='k',    ls ='-.',marker ='s',     ms =8,     label = Playe
plt.plot(Salary[9], c='k',    ls ='-.',marker ='o',     ms =8,     label = Playe




plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1))
plt.xticks(list(range(0,10)), Seasons,rotation = 'horizontal')
plt.show()
```


Sachin
Rahul
Smith
Sami
Pollard
Morris
Samson
Dhoni
Kohli
Sky
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019