Project 1

Ipl data analysis using numpy and matplotlib.

In [3]:

```
#Import numpy
import numpy as np

#Seasons
Seasons = ["2015","2016","2017","2018","2019","2020","2021","2022","2023","2024"]
Sdict = {"2015":0,"2016":1,"2017":2,"2018":3,"2019":4,"2020":5,"2021":6,"2022":7,"2023":8,"2024":9}

#Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","Kohli","Sky"]
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Samson":6,"Dhoni":7,"Kohli":8,"Sky":9}

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





Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Pollard_Salary, Morris_Salary, Samson_Salary, Dhoni_Salary, Kohli_Salary, Sky_Salary])




#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, Samson_G, Dhoni_G, Kohli_G, Sky_G])

#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, Morris_PTS, Samson_PTS, Dhoni_PTS, Kohli_PTS, Sky_PTS])
```

In [25]:

```
Salary
```

Out[25]:

```
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 [26]:

```
Games
```

Out[26]:

```
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 [27]:

```
Points
```

Out[27]:

```
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 [28]:

```
for idx, player in enumerate(Players):
    print(f"{idx}. {player}")
    print("Salaries:", Salary[idx])

# 💡 Explanation:
#  Salary[idx] accesses the entire row for that player (10 years of salary data).

#  Players[idx] gives the player's name.
```

```
0. Sachin
Salaries: [15946875 17718750 19490625 21262500 23034375 24806250 25244493 27849149
 30453805 23500000]
1. Rahul
Salaries: [12000000 12744189 13488377 14232567 14976754 16324500 18038573 19752645
 21466718 23180790]
2. Smith
Salaries: [ 4621800  5828090 13041250 14410581 15779912 14500000 16022500 17545000
 19067500 20644400]
3. Sami
Salaries: [ 3713640  4694041 13041250 14410581 15779912 17149243 18518574 19450000
 22407474 22458000]
4. Pollard
Salaries: [ 4493160  4806720  6061274 13758000 15202590 16647180 18091770 19536360
 20513178 21436271]
5. Morris
Salaries: [ 3348000  4235220 12455000 14410581 15779912 14500000 16022500 17545000
 19067500 20644400]
6. Samson
Salaries: [ 3144240  3380160  3615960  4574189 13520500 14940153 16359805 17779458
 18668431 20068563]
7. Dhoni
Salaries: [       0        0  4171200  4484040  4796880  6053663 15506632 16669630
 17832627 18995624]
8. Kohli
Salaries: [       0        0        0  4822800  5184480  5546160  6993708 16402500
 17632688 18862875]
9. Sky
Salaries: [ 3031920  3841443 13041250 14410581 15779912 14200000 15691000 17182000
 18673000 15000000]
```

In [29]:

```
Salary[0]
```

Out[29]:

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

In [30]:

```
Salary.min(0)
```

Out[30]:

```
array([       0,        0,        0,  4484040,  4796880,  5546160,
        6993708, 16402500, 17632688, 15000000])
```

In [31]:

```
Games[1:5]
```

Out[31]:

```
array([[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]])
```

In [32]:

```
Games[1,5]
```

Out[32]:

```
np.int64(72)
```

In [33]:

```
Salary/Games
```

Out[33]:
```
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 [34]:
```
np.round(Salary//Games)
```

Out[34]:
```
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 [35]:
```
import warnings
warnings.filterwarnings("ignore")
```

🔍 Purpose:

This disables warning messages in your script — helpful when you want to hide non-critical warnings (like deprecation or performance warnings) during data analysis or plotting.

⚠️ When to Use:

✅ Useful during exploratory data analysis (EDA) to keep the output clean.

❌ But avoid in production or final scripts — warnings are often helpful signals for bugs or upcoming issues.

In [ ]:
```

```

In [36]:
```
import matplotlib.pyplot as plt
```

In [37]:
```
Salary[0]
```

Out[37]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

✅ General Formula for plt.plot() (Recommended Order):

```
plt.plot(
    x,                     # X-axis data (e.g., index or Seasons)
    y,                     # Y-axis data (e.g., Salary[0])
    color='b',             # Line color (short: 'r', 'g', 'b', etc. or hex)
    linestyle='-',         # Line style ('-', '--', '-.', ':', or '')
    linewidth=2,           # Line thickness in points
    marker='o',            # Marker style (circle, star, triangle, etc.)
    markersize=6,          # Marker size in points
    label='Label Name',    # Label for legend
    alpha=1.0,             # Transparency (0.0 to 1.0)
    zorder=1               # Layer order: higher = on top
)
```

Attribute	Purpose
`x`, `y`	Data to plot
`color` / `c`	Line and marker color
`linestyle` / `ls`	Line pattern (solid, dashed, etc.)
`linewidth` / `lw`	Line thickness
`marker`	Marker shape (circle, triangle, etc.)
`markersize` / `ms`	Marker size in points
`label`	Name to display in the legend
`alpha`	Opacity (0 = transparent, 1 = opaque)
`zorder`	Drawing order (higher value = drawn above)

📌 Core Attributes of `plt.plot()`

Attribute	Description	Example
`x` / `y`	Data values (automatically uses index for x if not given)	`plt.plot(x, y)`
`c` or `color`	Line and marker color	`"red"`, `"b"`, `"#00ff00"`
`ls` or `linestyle`	Line style (`'-'`, `'--'`, `':'`, etc.)	`ls="--"`
`lw` or `linewidth`	Thickness of the line	`lw=2`
`marker`	Marker symbol (`'o'`, `'*'`, `'s'`, etc.)	`marker="*"`
`ms` or `markersize`	Size of the marker	`ms=10`
`label`	Label for legend	`label="Sachin"`
`alpha`	Transparency (0 to 1)	`alpha=0.6`
`zorder`	Stack order (higher = on top)	`zorder=3`

✏️ Common Colors (`c=` or `color=`)

Short Code	Color
`'b'`	Blue
`'g'`	Green
`'r'`	Red
`'c'`	Cyan
`'m'`	Magenta
`'y'`	Yellow
`'k'`	Black
`'w'`	White

📐 Common Line Styles (`ls=`)

Style	Symbol	Description
`'solid'`	`'-'`	Solid line
`'dashed'`	`'--'`	Dashed line
`'dashdot'`	`'-.'`	Dash-dot line
`'dotted'`	`':'`	Dotted line
`''` or `None`		No line

🔹 Common Marker Types (`marker=`)

Marker	Shape	Description
`'o'`	●	Circle
`'*'`	✱	Star
`'s'`	■	Square
`'D'`	♦	Diamond
`'+'`	+	Plus
`'x'`	x	X
`'<'`	◄	Left triangle
`'>'`	►	Right triangle
`'^'`	▲	Up triangle
`'v'`	▼	Down triangle

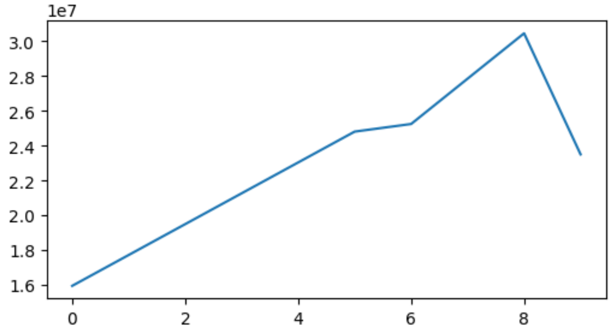
Marker	Shape	Description
`' '` or `None`		No marker

✅ Example Putting All Together:

```
plt.plot(Salary[0],
         color="orange",
         linestyle="--",
         linewidth=2,
         marker="D",
         markersize=8,
         label="Sachin's Salary",
         alpha=0.8)
plt.legend()
plt.grid(True)
plt.show()
```

In [38]:
```
plt.plot(Salary[0])
```

Out[38]:
```
[<matplotlib.lines.Line2D at 0x184d3048b50>]
```

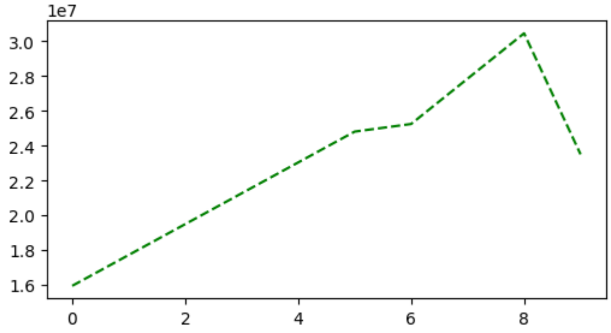


line style (ls)

`ls` Value	Description	Appearance
`"-"`	Solid line	————
`"--"`	Dashed line	— — — —
`"-."`	Dash-dot line	—·—·
`":"`	Dotted line	········
`""` or `None`	No line	Just markers

In [39]:
```
plt.plot(Salary[0],color="green",ls="--")
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x184d6a4b850>]
```

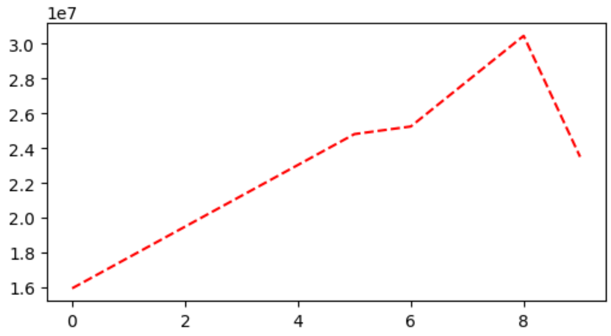


color (c)

`c` Value	Color
`"r"`	Red
`"g"`	Green
`"b"`	Blue
`"m"`	Magenta
`"c"`	Cyan
`"y"`	Yellow
`"k"`	Black
`"w"`	White
`"orange"`, `"purple"`	Named colors

In [40]:
```
plt.plot(Salary[0],  c="red"  ,ls="--")
```

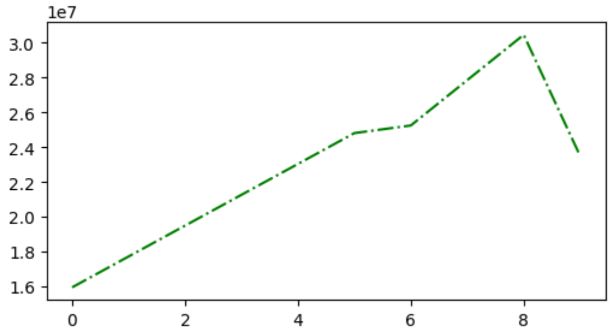
Out[40]:
```
[<matplotlib.lines.Line2D at 0x184d6ab9cd0>]
```



In [41]:
```
%matplotlib inline
plt.rcParams["figure.figsize"]=6,3
#You're setting up your Matplotlib environment for inline plotting and customizing figure size.
# 💡 Notes:
# %matplotlib inline is a magic command used only in Jupyter Notebooks. It displays plots inside the notebook.

# plt.rcParams["figure.figsize"] = (6, 3) sets a default plot size for all your plots.
```

In [42]:
```
plt.plot(Salary[0],  c="green"  ,ls="-.")
```

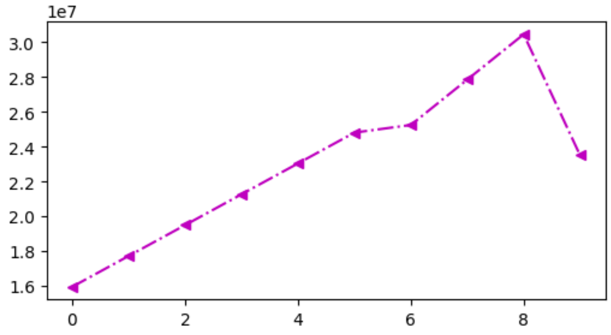
Out[42]:
```
[<matplotlib.lines.Line2D at 0x184d6b16090>]
```



Control Markers (marker)

Marker	Symbol	Description
`"o"`	●	Circle marker
`"s"`	■	Square marker
`"^"`	▲	Up triangle
`"v"`	▼	Down triangle
`"<"`	◄	Left triangle
`">"`	►	Right triangle
`"*"`	∗	Star
`"D"`	♦	Diamond
`"x"`	x	X marker
`"+"`	+	Plus marker
`""` or `None`	—	**No marker** (default)

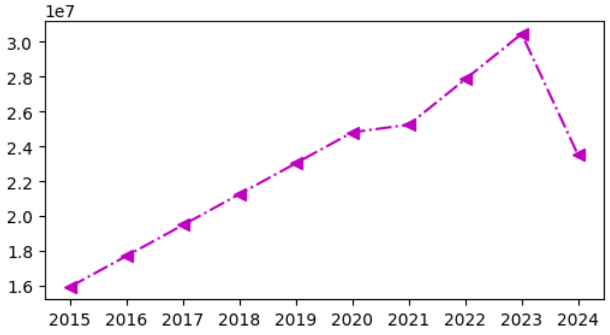
In [43]:
```
plt.plot(Salary[0],  c="m"  ,ls="-.",marker="<")
plt.show()
```



Parameter	Value	Meaning
`Salary[0]`	List	Salary data for Sachin
`c="m"`	Magenta	Line color
`ls="-."`	Dash-dot	Line style
`marker="<"`	Triangle	Marker style (left triangle)
`ms=7`	7 px	Marker size

In [44]:

```
plt.plot(Salary[0],c="m",ls="-.",marker="<",ms=7)
plt.xticks(list(range(10)),Seasons)
plt.show()
```



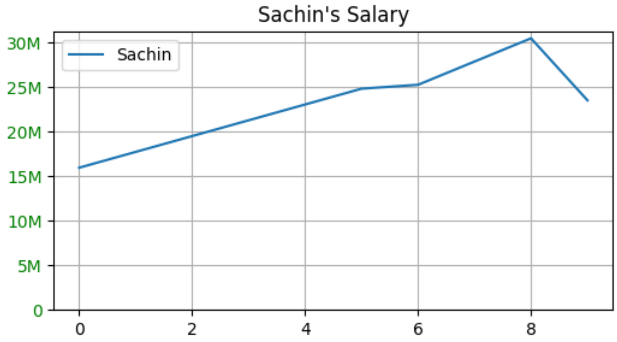
plt.xticks() and plt.xticks() General formula

```
plt.xticks(
    ticks=None,                 # Positions of ticks (list of numbers)
    labels=None,                # Custom labels for each tick (same length as ticks)
    rotation=0,                 # Rotation angle of labels (degrees)
    fontsize=None,              # Font size of labels
    color=None,                 # Color of tick labels
    horizontalalignment=None  # 'left', 'center', or 'right'
)
plt.yticks(
    ticks=None,                 # Positions of ticks (list of numbers)
    labels=None,                # Custom labels (same length as ticks)
    rotation=0,                 # Rotation angle of labels (in degrees)
    fontsize=None,              # Font size of labels
    color=None,                 # Color of tick labels
    horizontalalignment=None  # 'left', 'center', or 'right'
)
```

Parameter	Type	Default	Description	Common Values	Example
`ticks`	`list[int]` / array	`None`	Positions where ticks should appear	`range(10)`, `[0, 2, 4]`	`plt.xticks(ticks=[0, 2, 4, 6, 8])`
`labels`	`list[str]`	`None`	Custom labels for ticks (same length as `ticks`)	`["2015", "2016"]`, `['Low', 'High']`	`plt.xticks(ticks=[0,1], labels=["Low","High"])`
`rotation`	`int` / `float`	`0`	Rotates the tick labels in degrees	`0`, `45`, `90`, `270`	`plt.xticks(rotation=45)`
`fontsize`	`int` / `str`	`None`	Font size of tick labels	`10`, `'small'`, `'x-large'`	`plt.xticks(fontsize='large')`
`color`	`str`	`None`	Color of the tick label text	`'black'`, `'red'`, `'#336699'`	`plt.xticks(color='green')`
`horizontalalignment`	`str`	`None`	Alignment of tick labels horizontally	`'left'`, `'center'`, `'right'`	`plt.xticks(horizontalalignment='right')`

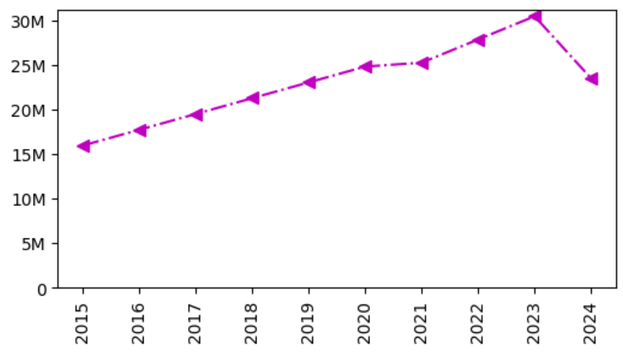
In [45]:

```
plt.plot(Salary[0], label="Sachin")

plt.yticks(
    ticks=range(0, 35000000, 5000000),  # Salary steps
    labels=["0", "5M", "10M", "15M", "20M", "25M", "30M"],
    rotation=0,
    fontsize=10,
    color='green'
)

plt.title("Sachin's Salary")
plt.legend()
plt.grid(True)
plt.show()
```



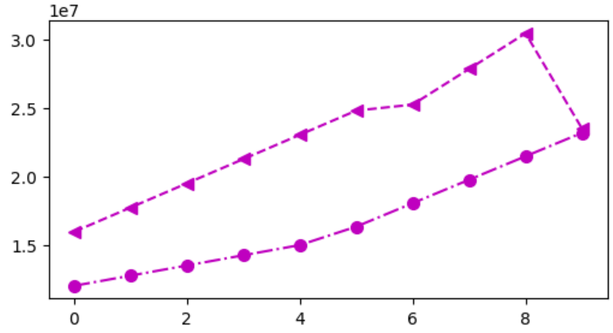
In [46]:

```
plt.plot(Salary[0],c="m",ls="-.",marker="<",ms=7)
plt.xticks(list(range(10)),Seasons,rotation="vertical")
plt.yticks(list(range(0,30000001,5000000)),["0", "5M", "10M", "15M", "20M", "25M", "30M"])
plt.show()
```



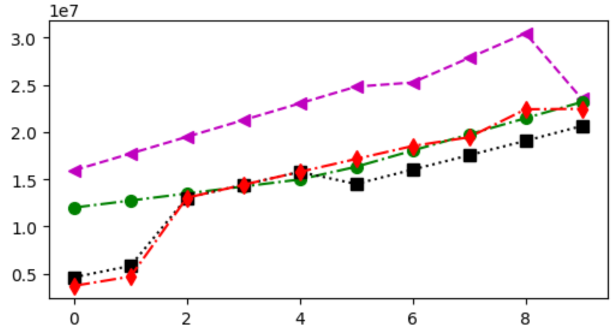
```
In [47]: plt.plot(Salary[0],c="m",ls="--",marker="<",ms=7)
         plt.plot(Salary[1],c="m",ls="-.",marker="o",ms=7)
```

```
Out[47]: [<matplotlib.lines.Line2D at 0x184d6c61e90>]
```



```
In [48]: plt.plot(Salary[0],c="m",ls="--",marker="<",ms=7,label=Players[0])
         plt.plot(Salary[1],c="g",ls="-.",marker="o",ms=7,label=Players[1])
         plt.plot(Salary[2],c="black",ls=":",marker="s",ms=7,label=Players[2])
         plt.plot(Salary[3],c="r",ls="-.",marker="d",ms=7,label=Players[3])
```

```
Out[48]: [<matplotlib.lines.Line2D at 0x184d6c35fd0>]
```



```
In [ ]:
```