In [4]:

```
import numpy as np
np.random.randn(5)
```

Out[4]:

```
array([-1.51188848,  1.64801055,  0.57945657, -0.02636647,  0.55151785])
```

In [5]:

```
# inp=s2b3c6
# out=ssbbbcccccc
```

In [10]:

```
# inp=s2b3c6
s=input()
out=""
for i in s:
    if i.isalpha():
        al=i
    else:
        dig=int(i)
        out=out+al*dig
print(out)
```

```
s2b3c6
ssbbbcccccc
```

In [11]:

```
import numpy as np
```

In [12]:

```
k=np.arange(10)
k
```

Out[12]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [13]:

```
k.reshape(5,2)
```

Out[13]:

```
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
```

In [15]:

```
k.dtype
```

Out[15]:

```
dtype('int64')
```

In [17]:

```
k.ndim
```

Out[17]:

```
1
```

In [20]:

```
kk=np.full((5,2),4)
kk
```

Out[20]:

```
array([[4, 4],
       [4, 4],
       [4, 4],
       [4, 4],
       [4, 4]])
```

In [21]:

```
rd=np.random.rand(5)
```

In [22]:

```
rd
```

Out[22]:

```
array([0.71068439, 0.40666771, 0.9709669 , 0.65945026, 0.20744038])
```

In [25]:

```
rd1=np.random.randn(5)
```

In [26]:

```
rd1
```

Out[26]:

```
array([ 1.82929781, -0.14368896,  0.01729401, -1.49260202, -0.36373607])
```

In [32]:

```
ktr=np.arange(12).reshape(6,2)
```

In [34]:

```
ktr
```

Out[34]:

```
array([[ 0,  1],
       [ 2,  3],
       [ 4,  5],
       [ 6,  7],
       [ 8,  9],
       [10, 11]])
```

In [40]:

```
ktr.min()
```

Out[40]:

```
0
```

In [41]:

```
ktr.max()
```

Out[41]:

```
11
```

In [42]:

```
ktr.std()
ktr.std()
```

Out[42]:

```
3.452052529534663
```

In [45]:

```
np.min(ktr,axis=0)
```

Out[45]:

```
array([0, 1])
```

In [46]:

```
np.max(ktr,axis=0)
```

Out[46]:

```
array([10, 11])
```

In [47]:

```
np.min(ktr,axis=1)
```

Out[47]:

```
array([ 0,  2,  4,  6,  8, 10])
```

In [48]:

```
np.max(ktr,axis=1)
```

Out[48]:

```
array([ 1,  3,  5,  7,  9, 11])
```

In [51]:

```
ln=np.linspace(1,100,5)
```

In [53]:

```
ln
```

Out[53]:

```
array([  1.  ,  25.75,  50.5 ,  75.25, 100.  ])
```

In [54]:

```
ksr=np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [55]:

```
ksr
```

Out[55]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [57]:

```
# Pandas is  lib from
import pandas as pd
```

In [59]:

```
da=pd.Series([1,2,3,4,5])
```

In [60]:

```
da
```

Out[60]:

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [61]:

```
slt=np.arange(5,10,1)
da2=pd.Series(slt)
```

In [62]:

```
da2
```

Out[62]:

```
0    5
1    6
2    7
3    8
4    9
dtype: int64
```

In [63]:

```
da2.values
```

Out[63]:

```
array([5, 6, 7, 8, 9])
```

In [64]:

```
da2.index
```

Out[64]:

```
RangeIndex(start=0, stop=5, step=1)
```

In [65]:

```
da.min()
```

Out[65]:

```
1
```

In [66]:

```
da.max()
```

Out[66]:

```
5
```

In [71]:

```
si=np.arange(5,10,1)
da3=pd.Series(si,index=["a","b","c","d","e"])
```

In [70]:

```
da3
```

Out[70]:

```
a    5
b    6
c    7
d    8
```

```
d    8
e    9
dtype: int64
```

In [72]:

```
da3[4]==da3["e"]
```

Out[72]:

```
True
```

In [75]:

```
da3[da3>7]
```

Out[75]:

```
d    8
e    9
dtype: int64
```

In [76]:

```
print(da3[4])
```

```
9
```

In [77]:

```
# Data Frames in Pandas
# tabler data rows and coluns
# 2 dim
import pandas as pd
```

In [80]:

```
ria={"name":["vidya","surya","swami","sdc","susma"],
     "dob":["07-05","29-11","17-05","01-05","19-03"],
     "branch":["cs","ml","ai","ds","cv"]
     }
df=pd.DataFrame(ria)
```

In [85]:

```
df.head(3)
```

Out[85]:

	name	dob	branch
0	vidya	07-05	cs
1	surya	29-11	ml
2	swami	17-05	ai

In [87]:

```
df.tail(3)
```

Out[87]:

	name	dob	branch
2	swami	17-05	ai
3	sdc	01-05	ds
4	susma	19-03	cv

In [90]:

```
df1=pd.DataFrame(df,columns=["name","dob","branch","mks"])
```

In [91]:

```
df1
```

Out[91]:

	name	dob	branch	mks
0	vidya	07-05	cs	NaN
1	surya	29-11	ml	NaN
2	swami	17-05	ai	NaN
3	sdc	01-05	ds	NaN
4	susma	19-03	cv	NaN

In [92]:

```
df1.isnull()
```

Out[92]:

	name	dob	branch	mks
0	False	False	False	True
1	False	False	False	True
2	False	False	False	True
3	False	False	False	True
4	False	False	False	True

In [93]:

```
df1["name"]
```

Out[93]:

```
0    vidya
1    surya
2    swami
3      sdc
4    susma
Name: name, dtype: object
```

In [94]:

```
df1["dob"]
```

Out[94]:

```
0    07-05
1    29-11
2    17-05
3    01-05
4    19-03
Name: dob, dtype: object
```

In [97]:

```
df1["mks"]=[12,13,14,15,16]    #np.random.randn(5)
```

In [98]:

```
df1
```

Out[98]:

	name	dob	branch	mks
0	vidya	07-05	cs	12
1	surya	29-11	ml	13
2	swami	17-05	ai	14
3	sdc	01-05	ds	15
4	susma	19-03	cv	16

In [100]:

```
df1.drop(4)
```

Out[100]:

	name	dob	branch	mks
0	vidya	07-05	cs	12
1	surya	29-11	ml	13
2	swami	17-05	ai	14
3	sdc	01-05	ds	15

In [104]:

```
df1.drop(4)
```

Out[104]:

	name	dob	branch	mks
0	vidya	07-05	cs	12
1	surya	29-11	ml	13
2	swami	17-05	ai	14
3	sdc	01-05	ds	15

In [105]:

```
ls
```

```
 23-06-2021.ipynb     class1.py   __pycache__/             'unnamed (2).png'
 24-06-2021.ipynb     data/        python-21-06-2021.ipynb   unnamed.png
 25-06-2021.ipynb     ksrcs.csv    unnamed1.jpg              untitled1.txt
 26-june-2020.ipynb   module.py  'unnamed (1).png'           untitled.txt
```

In [106]:

```
# read_csv()
# read_json()
# read_excel()
```

In [107]:

```
data=pd.read_csv("ksrcs.csv")
data
```

Out[107]:

	Roll-No	Name	Python	C	Java	DS	OS	DBMS	Marks	Grade
0	20JR1A0590	M.Sushma	4	97	4	88	41	52	286	Pass
1	20JR1A0591	Nagumothu Navya	80	77	49	45	70	0	321	Pass
2	20JR1A0592	N.Lakshmi Vasavi	43	69	21	8	93	91	325	Pass
3	20JR1A0593	Sushma Namburi	34	72	97	42	33	71	349	Pass
4	20JR1A0594	N.Lakshmi Chandana	93	95	14	10	73	20	305	Pass

	Roll-No	Name	Python	C	Java	DS	OS	DBMS	Marks	Grade
4	20JR1A0594	N.lakshmi Chandana	93	95	14	10	73	20	305	Pass
...	...	...	...	...	...	...	...	...	...	...
156	20JR1A4442	VIVEK	66	87	2	79	92	16	342	Pass
157	20JR1AO5J2	Charan Siva Sai	73	58	33	56	56	19	295	Pass
158	20jr1A0518	anusha	33	61	26	53	29	18	220	Pass
159	20jr1a0552	Abc	68	26	18	20	24	70	226	Pass
160	20jr1ao5d8	hello	16	94	98	9	58	10	285	Pass

161 rows × 10 columns

In [109]:

```
data.head(20)
```

Out[109]:

	Roll-No	Name	Python	C	Java	DS	OS	DBMS	Marks	Grade
0	20JR1A0590	M.Sushma	4	97	4	88	41	52	286	Pass
1	20JR1A0591	Nagumothu Navya	80	77	49	45	70	0	321	Pass
2	20JR1A0592	N.Lakshmi Vasavi	43	69	21	8	93	91	325	Pass
3	20JR1A0593	Sushma Namburi	34	72	97	42	33	71	349	Pass
4	20JR1A0594	N.lakshmi Chandana	93	95	14	10	73	20	305	Pass
5	20JR1A0595	swapna	70	22	37	41	1	16	187	Fail
6	20JR1A0596	GSP SRIKANTH	95	65	2	86	91	66	405	Pass
7	20JR1A0597	prakash	72	42	54	45	98	84	395	Pass
8	20JR1A0598	harsha Ravuri	86	23	82	27	77	4	299	Pass
9	20JR1A05A2	K Nagendra Babu	25	100	72	56	6	63	322	Pass
10	20JR1A05A3	NAVEEN	54	31	80	15	52	53	285	Pass
11	20JR1A05A5	K.SHOWRIBABU	46	91	32	83	65	2	319	Pass
12	20JR1A05A6	k.Prasanth Kumar	28	45	55	22	12	64	226	Pass
13	20JR1A05A8	Ansar	23	42	94	59	34	89	341	Pass
14	20JR1A05B0	K.SRAVAN	27	96	30	16	53	11	233	Pass
15	20JR1A05B1	RAHUL_KITS	2	44	98	54	59	43	300	Pass
16	20JR1A05B2	SIDDHARDHA REDDY	99	69	8	43	10	86	315	Pass
17	20JR1A05B3	L.GOPI	31	6	1	33	9	30	110	Fail
18	20JR1A05B4	M.Anil	38	1	21	89	9	39	197	Fail
19	20JR1A05B5	MAHANKALI SAI	8	2	13	74	96	28	221	Pass

In [110]:

```
data.tail(10)
```

Out[110]:

	Roll-No	Name	Python	C	Java	DS	OS	DBMS	Marks	Grade
151	20JR1A4362	SK.B.SHOHEB AKTHAR	81	95	27	27	34	88	352	Pass
152	20JR1A4363	MUNEER	46	57	55	9	70	50	287	Pass
153	20JR1A4364	Naga Sai Durgesh Singamsetty	65	62	9	54	46	67	303	Pass
154	20JR1A4431	Sakhamuri. Lathasri	65	89	57	32	82	80	405	Pass
155	20JR1A4435	Rushitha	17	61	83	12	3	67	243	Pass
156	20JR1A4442	VIVEK	66	87	2	79	92	16	342	Pass

	Roll-No	Name	Python	C	Java	DS	OS	DBMS	Marks	Grade
157	20JR1AO5J2	Charan Siva Sai	73	58	33	56	56	19	295	Pass
158	20jr1A0518	anusha	33	61	26	53	29	18	220	Pass
159	20jr1a0552	Abc	68	26	18	20	24	70	226	Pass
160	20jr1ao5d8	hello	16	94	98	9	58	10	285	Pass

In [117]:

```
data.min()
```

Out[117]:

```
Roll-No    20JR1A0590
Python              0
C                   0
Java                0
DS                  1
OS                  0
DBMS                0
Marks             107
Grade            Fail
dtype: object
```

In [118]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 161 entries, 0 to 160
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Roll-No  161 non-null    object
 1   Name     154 non-null    object
 2   Python   161 non-null    int64 
 3   C        161 non-null    int64 
 4   Java     161 non-null    int64 
 5   DS       161 non-null    int64 
 6   OS       161 non-null    int64 
 7   DBMS     161 non-null    int64 
 8   Marks    161 non-null    int64 
 9   Grade    161 non-null    object
dtypes: int64(7), object(3)
memory usage: 12.7+ KB
```

In [119]:

```
data.describe()
```

Out[119]:

	Python	C	Java	DS	OS	DBMS	Marks
count	161.000000	161.000000	161.000000	161.000000	161.000000	161.000000	161.000000
mean	51.664596	48.503106	45.776398	42.273292	52.093168	49.043478	289.354037
std	29.337677	29.097277	30.168894	28.250882	29.633132	29.139181	67.901161
min	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	107.000000
25%	27.000000	23.000000	20.000000	17.000000	29.000000	26.000000	234.000000
50%	50.000000	47.000000	49.000000	40.000000	52.000000	49.000000	295.000000
75%	80.000000	70.000000	69.000000	65.000000	79.000000	75.000000	331.000000
max	99.000000	100.000000	100.000000	100.000000	100.000000	100.000000	441.000000

In [123]:

```
c=data["C"]
c
```

Out[123]:

```
0      97
1      77
2      69
3      72
4      95
       ..
156    87
157    58
158    61
159    26
160    94
Name: C, Length: 161, dtype: int64
```

In [126]:

```
ds=data["DS"]
ds
```

Out[126]:

```
0      88
1      45
2       8
3      42
4      10
       ..
156    79
157    56
158    53
159    20
160     9
Name: DS, Length: 161, dtype: int64
```

In [128]:

```
py=data["Python"]
py
```

Out[128]:

```
0       4
1      80
2      43
3      34
4      93
       ..
156    66
157    73
158    33
159    68
160    16
Name: Python, Length: 161, dtype: int64
```

In [130]:

```
Ja=data["Java"]
Ja
```

Out[130]:

```
0       4
1      49
2      21
3      97
4      14
       ..
156     2
157    33
158    26
159    18
160    98
```

```
Name: Java, Length: 161, dtype: int64
```

In [131]:

```
db=data["DBMS"]
```

In [132]:

```
os=data["OS"]
```

In [133]:

```
da=data["DS"]
```

In [143]:

```
kk=data["DS"].drop(3)
kk
```

Out[143]:

```
0      88
1      45
2       8
4      10
5      41
       ..
156    79
157    56
158    53
159    20
160     9
Name: DS, Length: 160, dtype: int64
```

In [139]:

```
data.drop('Python',
  axis='columns', inplace=True)
```

In [147]:

```
data.drop('C',axis=1,inplace=True)
```

In [141]:

```
data
```

Out[141]:

	Roll-No	Name	C	Java	DS	OS	DBMS	Marks	Grade
0	20JR1A0590	M.Sushma	97	4	88	41	52	286	Pass
1	20JR1A0591	Nagumothu Navya	77	49	45	70	0	321	Pass
2	20JR1A0592	N.Lakshmi Vasavi	69	21	8	93	91	325	Pass
3	20JR1A0593	Sushma Namburi	72	97	42	33	71	349	Pass
4	20JR1A0594	N.lakshmi Chandana	95	14	10	73	20	305	Pass
...	...	...	...	...	...	...	...	...	...
156	20JR1A4442	VIVEK	87	2	79	92	16	342	Pass
157	20JR1AO5J2	Charan Siva Sai	58	33	56	56	19	295	Pass
158	20jr1A0518	anusha	61	26	53	29	18	220	Pass
159	20jr1a0552	Abc	26	18	20	24	70	226	Pass
160	20jr1ao5d8	hello	94	98	9	58	10	285	Pass

161 rows × 9 columns

In [140]:

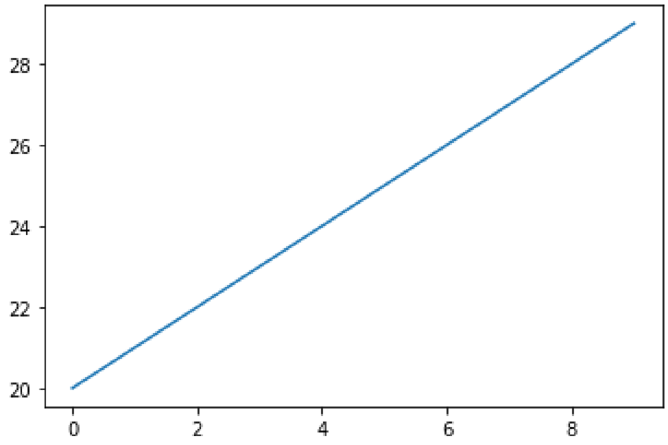
```
import matplotlib.pyplot as plt
# pie
# bar
# hist
# plot
# scattor
```

In [153]:

```
x=np.arange(10)
y=np.arange(20,40,2)
```

In [151]:

```
plt.plot(x,y)
plt.show()
```

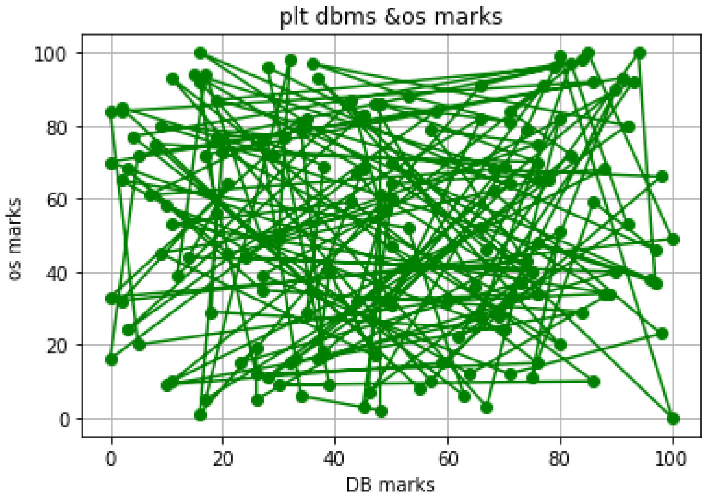


In [160]:

```
#dir(plt)
```

In [169]:

```
plt.plot(db,os,c="g",marker="o")
plt.grid()
plt.xlabel("DB marks")
plt.ylabel("os marks")
plt.title("plt dbms &os marks")
plt.savefig("first.jpg")
```



In [170]:

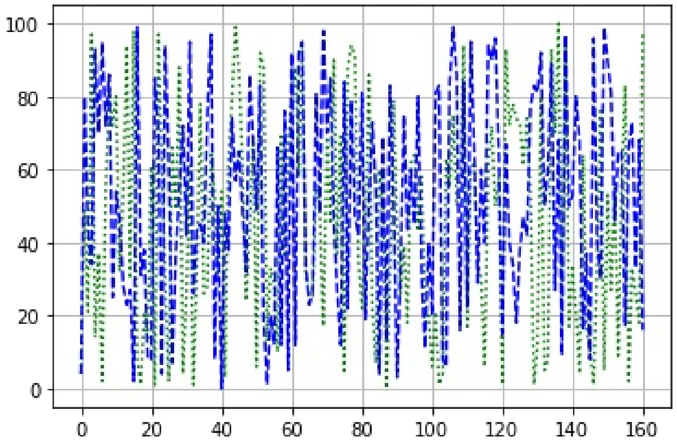
```
ls
```

```
 23-06-2021.ipynb      data/          python-21-06-2021.ipynb   untitled1.txt
 24-06-2021.ipynb      first.jpg      unnamed1.jpg              untitled.txt
```

```
25-06-2021.ipynb     ksrcs.csv      'unnamed (1).png'
26-june-2020.ipynb   module.py      'unnamed (2).png'
class1.py            __pycache__/    unnamed.png
```

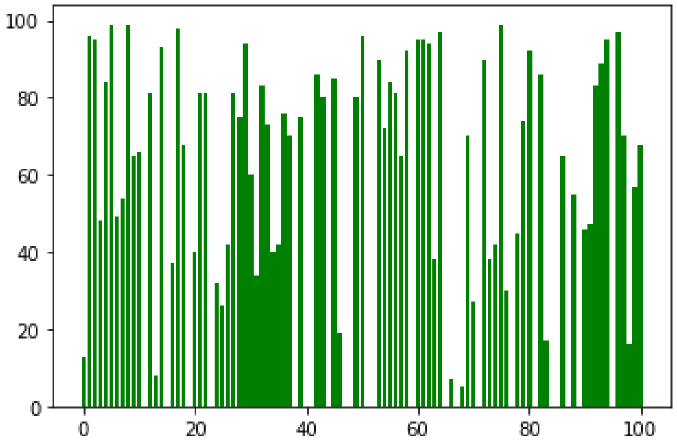
In [181]:

```
plt.plot(Ja,linestyle="dotted",c="g")
plt.plot(py,linestyle="dashed",c="b")
plt.grid()
plt.show()
```



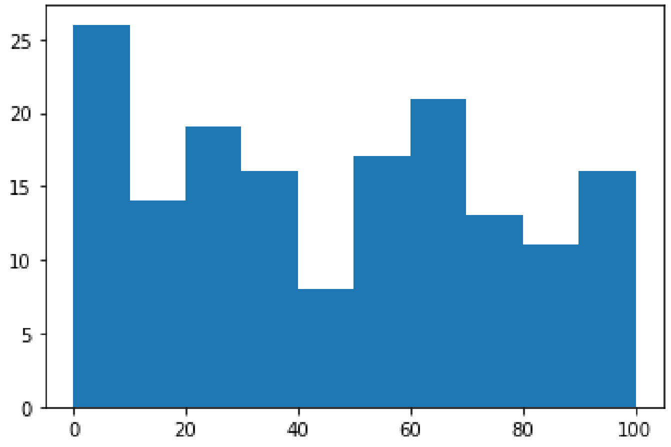
In [189]:

```
plt.bar(Ja,py,linestyle="dashed",color="g")
plt.show()
```



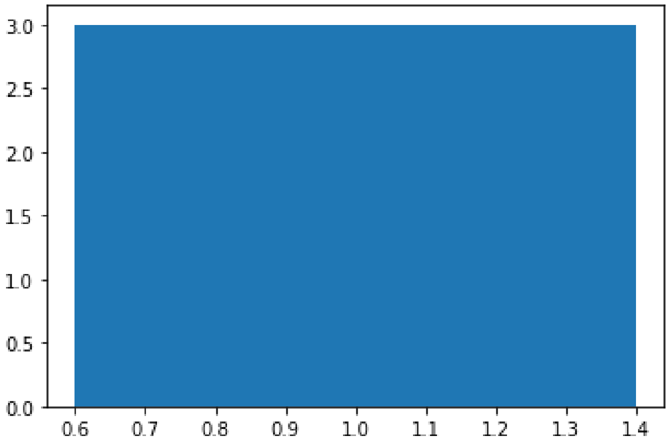
In [185]:

```
plt.hist(Ja)
plt.show()
```



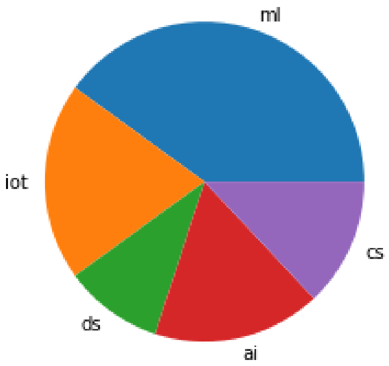
In [190]:

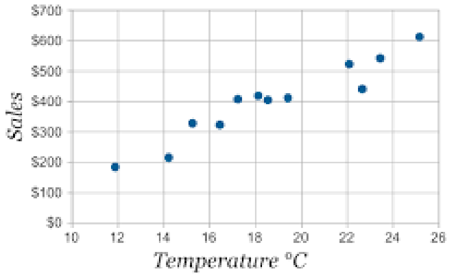
```
plt.bar(1,3)
plt.show()
```



In [195]:

```
course=["ml","iot","ds","ai","cs"]
per=[40,20,10,17,13]
plt.pie(per,labels=course)
plt.show()
```





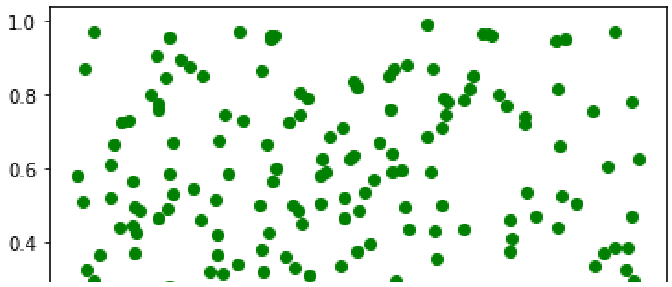
In [203]:

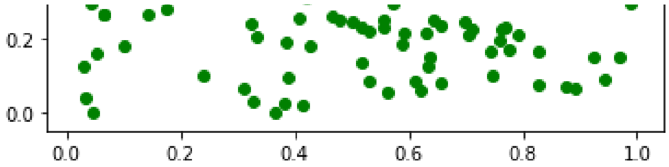
```
x=np.random.rand(200)
y=np.random.rand(200)
print(x)
print(y)
```

```
[0.94323873 0.2178483  0.44694767 0.19025008 0.13393941 0.55436078
 0.93323776 0.87313228 0.27230086 0.93374705 0.9988765  0.05234511
 0.71268967 0.70233545 0.21839467 0.42358198 0.43873305 0.78672617
 0.82174898 0.6852927  0.818201   0.23022014 0.67520298 0.65540422
 0.74102182 0.10965444 0.26238866 0.18264217 0.72658877 0.33013135
 0.6977536  0.00375571 0.49703552 0.85455337 0.95929284 0.75309718
 0.82171163 0.63065041 0.77198046 0.60230806 0.54884021 0.40412632
 0.21759743 0.10455019 0.77440018 0.73405155 0.92867501 0.69480377
 0.99773494 0.68174567 0.45850315 0.54508274 0.66024997 0.09282746
 0.53719172 0.52356532 0.77196457 0.65211336 0.77257107 0.07277951
 0.67233825 0.84033122 0.08805007 0.072106   0.95808629 0.5654013
 0.94925242 0.69166578 0.20382123 0.77265976 0.39051664 0.65467375
 0.32443924 0.23268035 0.74699028 0.63825175 0.76141822 0.80924766
 0.22914244 0.69036817 0.53812051 0.85656393 0.19486727 0.20841869
 0.82551986 0.63786642 0.47563829 0.89687261 0.45427762 0.41461314
 0.44581772 0.0312001  0.42625851 0.99484843 0.1227926  0.42043366
 0.96234321 0.91458331 0.09740364 0.23658641 0.28473949 0.02459766
 0.18850162 0.52415631 0.640989   0.26583627 0.3306538  0.80617409
 0.31366963 0.40678913 0.40068805 0.88472651 0.37537005 0.4759698
 0.331082   0.89094011 0.42841255 0.6139415  0.95331738 0.15693862
 0.85967297 0.02071279 0.09586203 0.48241024 0.56344858 0.25371849
 0.49440599 0.01641946 0.78942729 0.59273624 0.80674253 0.04014964
 0.44262    0.62234515 0.69491266 0.19206759 0.82518997 0.59820017
 0.19140145 0.73222779 0.75054112 0.26173907 0.45864447 0.47794874
 0.83729338 0.96523686 0.88724931 0.90067776 0.43120057 0.21097236
 0.24047394 0.34274697 0.64326798 0.22877578 0.06999505 0.06972563
 0.51079791 0.06599658 0.60638237 0.34725588 0.29342095 0.27469419
 0.89418993 0.63664139 0.13833864 0.11004531 0.84710318 0.0098371
 0.07395705 0.53282564 0.23909508 0.95039066 0.44855464 0.44847116
 0.03619178 0.76830815 0.11487262 0.5535124  0.30055669 0.514006
 0.67839538 0.97560446 0.2423567  0.25665309 0.87787273 0.01578369
 0.96628096 0.41223369 0.37105119 0.67449365 0.18876605 0.20421857
 0.79324698 0.92375577 0.57320954 0.9346173  0.08042362 0.77923006
 0.63569276 0.86445714]
[0.1973892  0.02113461 0.70418929 0.19438166 0.78717433 0.34159077
 0.64630276 0.3019619  0.67367579 0.09987805 0.75334482 0.35606159
 0.50884691 0.83663629 0.03341537 0.43635627 0.50085704 0.58471945
 0.86934005 0.9877864  0.10972835 0.56741456 0.83202598 0.50624234
 0.65110326 0.54202574 0.04100968 0.65664849 0.16711043 0.45812544
 0.67433462 0.76129708 0.28731439 0.81464076 0.15762017 0.45432985
 0.64748561 0.95007539 0.88505889 0.16217042 0.33146554 0.41546401
 0.96268794 0.0041822  0.68968536 0.6719196  0.00200193 0.04560791
 0.90243465 0.81659816 0.16869475 0.67152599 0.85610763 0.48057293
 0.12316847 0.07611453 0.82294536 0.17356566 0.19073853 0.7805826
 0.68771043 0.35479165 0.58990592 0.15150465 0.30923392 0.0691096
 0.72717195 0.92595771 0.77985389 0.05277    0.28643038 0.79875209
 0.03098234 0.89001037 0.91471492 0.32988927 0.67156638 0.97182147
 0.55284597 0.60369519 0.30702141 0.29035661 0.64363838 0.44190011
 0.10237877 0.00832087 0.44609269 0.94725263 0.68685256 0.70950419
 0.90844546 0.75488157 0.4634135  0.16397567 0.60925209 0.55899035
 0.6808228  0.64760428 0.89082587 0.32714361 0.42897756 0.39661681
 0.59315823 0.4720453  0.38514709 0.59296821 0.00762506 0.17438245
 0.45245985 0.13445796 0.56679553 0.17752662 0.6476247  0.20937797
 0.82544062 0.57697782 0.21887988 0.25956468 0.8133535  0.17669662
 0.448508   0.95686317 0.42949202 0.45588192 0.38438872 0.92187468
 0.81790797 0.52263864 0.9430222  0.53396862 0.97202964 0.57574369
 0.06709921 0.41653077 0.24737236 0.81116479 0.74394613 0.17238963
 0.29275886 0.95769246 0.83847832 0.08358577 0.3322544  0.11750931
 0.05231743 0.28949838 0.94828739 0.97330363 0.35994653 0.15260144
 0.64417912 0.05283058 0.6474903  0.57922263 0.5706082  0.82588487
 0.21482381 0.43450064 0.52611225 0.98531585 0.80740301 0.19116056
 0.8909832  0.93068572 0.36092874 0.13286341 0.01809021 0.93906804
 0.52450408 0.72879678 0.0465262  0.27463192 0.62244301 0.38116199
 0.92986361 0.84456837 0.81363241 0.3493645  0.90276046 0.49244166
 0.97198054 0.2677161  0.56645077 0.10077203 0.57772284 0.33214835
 0.0931356  0.79226917 0.07796339 0.48771934 0.67267989 0.97808144
 0.56724775 0.69726958 0.16995283 0.43491628 0.04674855 0.33801616
 0.16301006 0.6778629 ]
```

In [202]:

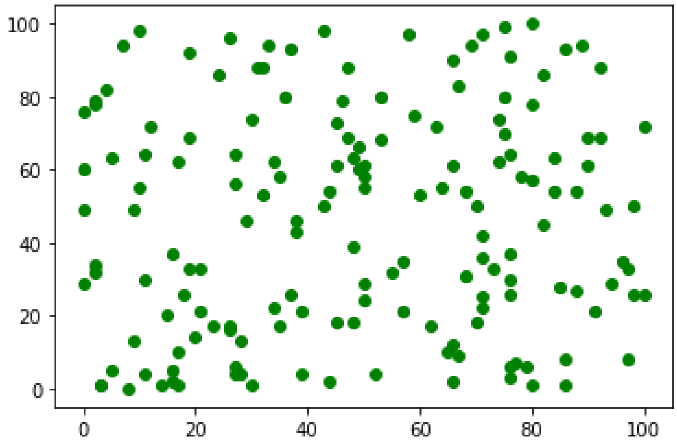
```
plt.scatter(x,y,c="g")
plt.show()
```





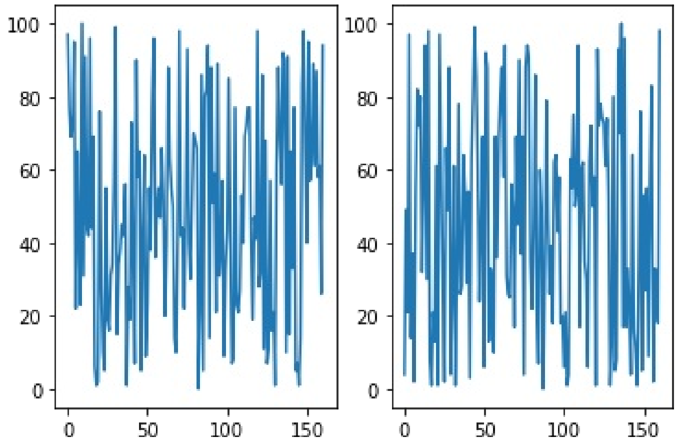
In [204]:

```
plt.scatter(db,Ja,c="g")
plt.show()
```



In [210]:

```
plt.subplot(1,2,1)
plt.plot(c)
plt.subplot(1,2,2)
plt.plot(Ja)
plt.show()
```



Python iterators

generators

In [211]:

```
l=[1,4,5,6,7]
for i in l:
    print(i)
```

```
1
4
5
6
7
```

In [216]:

```
l=[1,4,5,6,7]
da=iter(l)
print(next(da))
print(next(da))
print(next(da))
print(next(da))
print(next(da))
```

```
1
4
5
6
7
```

In [217]:

```
def gen():
    yield 10
    yield 20
    yield 30
    yield 40
    yield 50
h=gen()
for i in h:
    print(i)
```

```
10
20
30
40
50
```

In [ ]:

```

```