```
import numpy as np
x=np.array([3,4,4,5,6])
print(x)
```

```
import numpy as np
x=np.array([[1,2,3,4],[4,5,6,7]])
print(x)
y=np.dtype('i4')
```

```
import numpy as np
x=np.array([[1,2,3,4],
        [4,5,6,7],
        [7,8,9,4]])
print(x.shape)#it gives the shape of given array
y=x.reshape(4,3)#it changes the shape of array
print(y)
z=np.zeros(8)#it gives the array of zeros with 8 values
print(z)
p=np.zeros((3,3))#it gives the array of 3 by 3 of zero values
print(p)
q=np.ones(4)#it gives the array with value 1
print(q)
r=np.ones((3,3))#it gives array of 3 by 3 with value 1
print(r)
```

```
import numpy as np
x=[1,2,3,4]
a=np.asarray(x)
print(a)
b=np.asarray(x,dtype=float)
print(b)
y=[(2,5,8),(3,6,9)]
c=np.asarray(y)
print(c)
```

```
import numpy as np
a=np.arange(10)
print(a)
```

```
b=np.arange(2,10)
print(b)
c=np.arange(2,10,2)
print(c)
d=np.arange(5,51,5)
e=d.reshape((2,5))
print(e)
f=np.arange(20)
g=slice(2,18,2)
print(f[g])
h=f[2:18:2]
print(h)
```

```
import numpy as np
a=np.array([[2,5,8],[7,3,6],[9,4,3]])
print(a)
print(a[1:])
print(a[...,1])
print(a[1,...])
print(a[...,1:]
```

```
import numpy as np
a=np.array([[2,5],[8,5],[4,5]])
y=a[[0,1,2],[0,1,0]]#row,column
print(y)
```

```
import numpy as np
x=np.array([[3,4,5],[4,5,6],[7,8,9],[1,4,7]])
r=np.array([[0,0],[2,2]])
c=np.array([[0,2],[1,2]])
y=x[r,c]
print(y)
p=x[1:4,1:3]#row from 1 t0 3 and col 1 t0 2
print(p)
s=x[1:4,[2]]#row from 1 to 3 and col 2
print(s)
print(x[x>5])
```

```
import numpy as np#broadcasting
x=np.array([[3,4,5],[4,5,9],[8,9,11]])
y=np.array([2,8,9])
print(x*y)
print(x+y)
```

```
import numpy as np
a=np.arange(0,60,5)
b=a.reshape(3,4)
print(b)
for x in np.nditer(b):
   print(x)
print(b.T)#transpose of matrix
```

```
import numpy as np
a=np.arange(0,60,5)
b=a.reshape(3,4)
c=np.arange(5,65,5)
d=c.reshape(3,4)
print(np.add(b,d))
print(np.subtract(b,d))
print(np.multiply(b,d))
print(np.divide(b,d))
p=np.array[2,8,13,14]
print(np.power(a,2))
```

```
import numpy as np
a=np.array([2,8,13,14])
b=np.array([3,4,5,6])
print(np.mod(a,2))
print(np.power(a,2))
print(np.power(a,b))
```
