```
import numpy as np


#Question-1: program to create a numpy array starting from 2 till 50 with a stepsize of 3.
s=print(np.arange(2,50,3))
```

```
[ 2  5  8 11 14 17 20 23 26 29 32 35 38 41 44 47]
```

```
#Question-2: program to convert lists to numpy arrays and concatenate and sort.
l1=[7,8,9,10,11]
l2=[2,5,6,4,1]
arr1=np.array(l1)
print(arr1)
arr2=np.array(l2)
print(arr2)
print()
c=np.concatenate([arr1,arr2])
print(c)
s=np.sort(c)
print(s)
```

```
[ 7  8  9 10 11]
[2 5 6 4 1]

[ 7  8  9 10 11  2  5  6  4  1]
[ 1  2  4  5  6  7  8  9 10 11]
```

```
#Question-3: code to find the dimmensions of a ndarray and its size.
arr1=np.arange(10)
print(arr1)
print(np.ndim(arr1))
print(np.size(arr1))
print()
arr2=np.arange(10).reshape(2,5)
print(arr2)
print(np.ndim(arr2))
print(np.size(arr2))
```

```
[0 1 2 3 4 5 6 7 8 9]
1
10

[[0 1 2 3 4]
 [5 6 7 8 9]]
2
10
```

```
#Question-4: how to convert a 1D array into a 2D array.
```

```
#1.Using newaxis and expand_dims()
#numpy.newaxis is used to increase the dimmension of the existing array by one more dimmensio
#expand_dims() function is used to expand the shape of an array.
a=np.arange(6)
print(a)
e=a[:,np.newaxis]
np.expand_dims(a,0)
print(e)
print(e.ndim)
```

```
[0 1 2 3 4 5]
[[0]
 [1]
 [2]
 [3]
 [4]
 [5]]
2
```

```
#Question-5: consider two square numpy arrays.Stack them vertically and horizontally.
arr1=np.array([1,4,5,7,3])
arr2=np.array([5,2,3,5,6])
s1=np.square(arr1)
s2=np.square(arr2)
v=np.vstack((s1,s2))
print("Vertical stack:",v)
print()
h=np.hstack((s1,s2))
print("horizontal stack:",h)
```

```
Vertical stack: [[ 1 16 25 49  9]
 [25  4  9 25 36]]

horizontal stack: [ 1 16 25 49  9 25  4  9 25 36]
```

```
#Question-6: How to get unique items and counts of unique items?
arr=([1,2,3,4,5,5,2,3,2,2,1,7,8,9,9])
print(arr)
uniquevalues,indices,count=np.unique(arr,return_index=True,return_counts=True)
print("Uniquevalues:",uniquevalues)
print("indices:",indices)
print("counts:",count)
```

```
[1, 2, 3, 4, 5, 5, 2, 3, 2, 2, 1, 7, 8, 9, 9]
Uniquevalue: [1 2 3 4 5 7 8 9]
indices: [ 0  1  2  3  4 11 12 13]
counts: [2 4 2 1 2 1 1 2]
```