Excersise - 1

Program - 1

Create a code for stacking 1D arrays horizontally Given: Array-1: [1, 2, 3] Array-2: [4, 5, 6]

In [2]:
```
import numpy

ary_1 = numpy.array([1, 2, 3])
print()
ary_2 = numpy.array([4, 5, 6])
final_ary = numpy.hstack((ary_1, ary_2))

print(final_ary)
```

```
[1 2 3 4 5 6]
```

Program - 2

Create a program to get the unique values from the array with the aid of Numpy! Given: [1, 2, 2, 4, 3, 6, 4, 8]

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

a = [1, 2, 2, 4, 3, 6, 4, 8]

uniq_val = np.unique(a)
print(uniq_val)
```

```
[1 2 3 4 6 8]
```

Program - 3

Given Matrix a = [[1, 2, 3], [4, 9, 6], [7, 8, 9]]. Find its inverse using "linalg.inv" function in NumPy.

In [4]:
```
a = [[1, 2, 3], [4, 9, 6], [7, 8, 9]]

a_inv = np.linalg.inv(a)
print(a_inv)
```

```
[[-0.6875     -0.125       0.3125    ]
 [-0.125       0.25       -0.125     ]
 [ 0.64583333 -0.125      -0.02083333]]
```

Program - 4

Given 1D array a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Reshape it into a 3D array. Again reshape back to 1D array. Display all!

In [5]:
```
import numpy as np

a = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16])
a_3d = a.reshape(2, 4, 2)
a_1d_again=a_3d.reshape(-1)
print(a,"\n")
print(a_3d,"\n")
print(a_1d_again,"\n")
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]

[[[ 1  2]
  [ 3  4]
  [ 5  6]
  [ 7  8]]

 [[ 9 10]
  [11 12]
  [13 14]
  [15 16]]]

[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]
```

Program - 5

Suppose a sensor generates the following data Sequence as x = [2, 3, 5, 6, 7, 9]. If the Central/Observed Value is 7, calculate deviation of each data values.

In [6]:
```
import numpy as np

central_value = 7
x = np.array([2, 3, 5, 6, 7, 9])
deviation = x - central_value

print(deviation)
```

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

Program - 6

Create a 2D array [[0,1,7,0,0],[3,0,0,2,19]]. Count number of non-zero elements.

In [7]:
```
import numpy as np

array_2d = np.array([[0, 1, 7, 0, 0], [3, 0, 0, 2, 19]])
non_zero_count = np.count_nonzero(array_2d)

print(non_zero_count)
```

```
5
```

Program - 7

Create a list, num_list = [10, 20, 30, 40, 50]. Using np.random.choice, select the elements with probability values, 0.1, 0.3, 0, 0.4, 0.2.

In [8]:
```
import numpy as np

num_list = [10, 20, 30, 40, 50]
prob_values = [0.1, 0.3, 0, 0.4, 0.2]
selected_elements = np.random.choice(num_list, size=len(num_list), p=prob_values)

print(selected_elements)
```

```
[10 20 20 20 40]
```

Program - 8

Generate a 3x3 matrix with random values. Find its Transpose, Inverse Matrices.

In [9]:
```
import numpy as np

matrix = np.random.rand(3,3)

transpose_matrix = matrix.T

inverse_matrix = np.linalg.inv(matrix)

print("Original Matrix:\n", matrix)
print("Transpose Matrix:\n", transpose_matrix)
print("Inverse Matrix:\n", inverse_matrix)
```

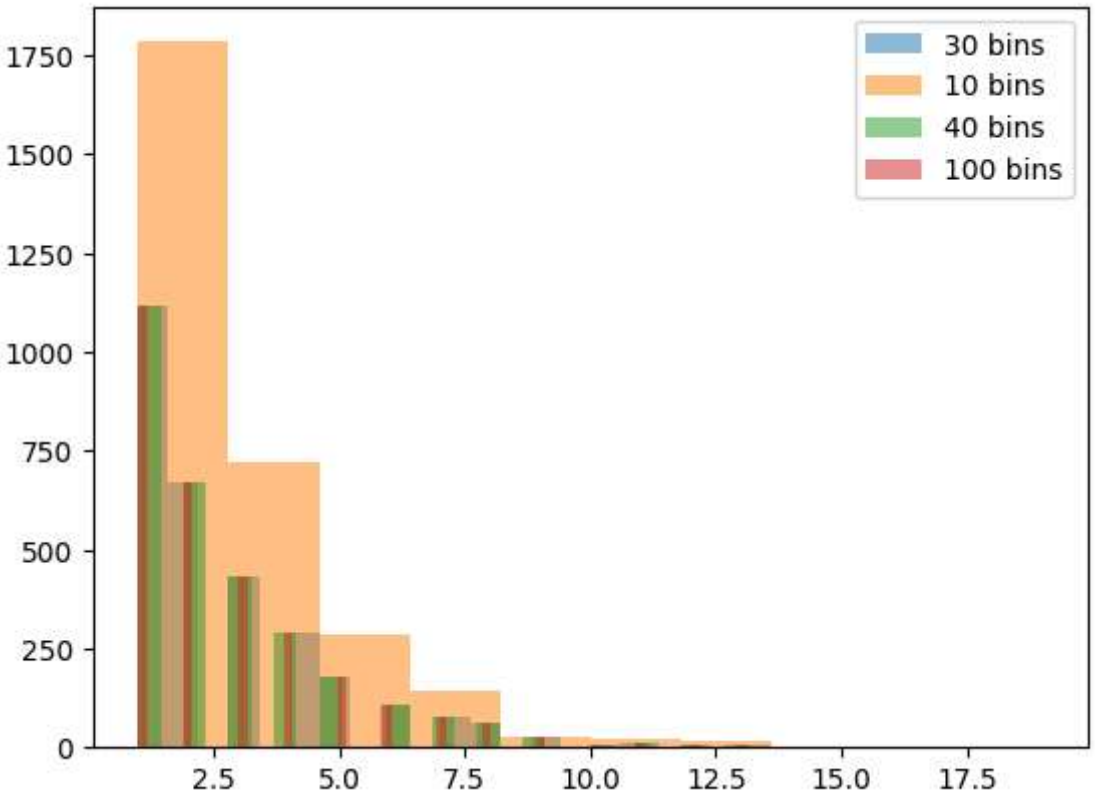
```
Original Matrix:
 [[0.70227817 0.93568614 0.94376383]
 [0.95120834 0.88089329 0.76154581]
 [0.49354494 0.69836903 0.92426588]]
Transpose Matrix:
 [[0.70227817 0.95120834 0.49354494]
 [0.93568614 0.88089329 0.69836903]
 [0.94376383 0.76154581 0.92426588]]
Inverse Matrix:
 [[-5.03858213  3.67137336  2.11985819]
 [ 8.98202169 -3.2711726  -6.4762311 ]
 [-4.09622285  0.51121424  4.84336165]]
```

Program - 9

Use Geometric Random distribution for generating 3000 samples with the probability of an individual success equal to 0.35. Plot the histogram with 30 bins. Observe changes in the plot for 10 / 40 / 100 bins.

In [10]:
```
import numpy as np
import matplotlib.pyplot as plt

samples = np.random.geometric(p=0.35, size=3000)

plt.hist(samples, bins=30, alpha=0.5, label='30 bins')
```

```
plt.hist(samples, bins=10, alpha=0.5, label='10 bins')
plt.hist(samples, bins=40, alpha=0.5, label='40 bins')
plt.hist(samples, bins=100, alpha=0.5, label='100 bins')

plt.legend()
plt.show()
```

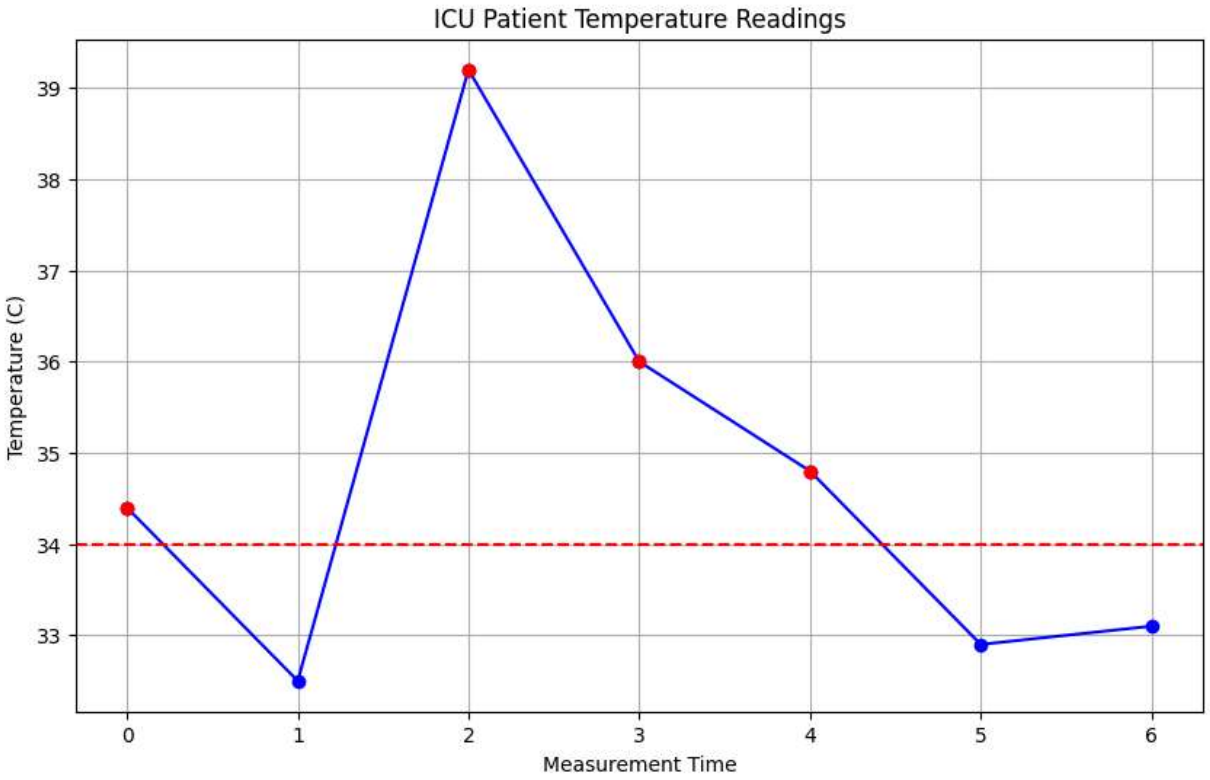


Program - 10

Suppose a temperature sensor of an ICU patient generates x=[34.4, 32.5, 39.2, 36.0, 34.8, 32.9, 33.1], Assume that <=34.0 C as normal, count number of times, the patient suffer by high temperature. Plot this temperature readings as a graph with proper title, x-axis and y-axis labels.

In [11]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.array([34.4, 32.5, 39.2, 36.0, 34.8, 32.9, 33.1])
high_temp_count = np.sum(x > 34.0)
plt.figure(figsize=(10, 6))
plt.plot(x, marker='o', linestyle='-', color='b')
plt.axhline(y=34.0, color='r', linestyle='--')

for i, temp in enumerate(x):
    if temp > 34.0:
        plt.plot(i, temp, marker='o', color='r')

plt.title('ICU Patient Temperature Readings')
plt.xlabel('Measurement Time')
```

```
plt.ylabel('Temperature (C)')
plt.grid(True)
plt.show()

print(f"Number of times the patient had a high temperature: {high_temp_count}")
```




```
Number of times the patient had a high temperature: 4
```