In [1]:

```
import numpy as np
from scipy import stats
import statistics
import pandas as pd
from scipy.stats import skew
from scipy.stats import kurtosis
import matplotlib.pyplot as plt
```

In [4]:

```
# Determine the central tendency of the below Population:
# [-993, -23,18,-2,-6,98,45,32,-45,843,1024,-256]
data = [-993, -23,18,-2,-6,98,45,32,-45,843,1024,-256]
a = np.mean(data)
b = np.median(data)
c = stats.mode(data)
print(f'Mean: {a}')
print(f'Median: {b}')
print(f'Mode: {c}')
```

```
Mean: 61.25
Median: 8.0
Mode: ModeResult(mode=array([-993]), count=array([1]))
```

In [7]:

```
# Calculate the Standard Deviation and Variance of the below sample :
# [-99, -2,18,-23,-61,1,982,45,32,-45]
s1=[-99, -2,18,-23,-61,1,982,45,32,-45]
stddev=statistics.stdev(s1)
var=statistics.variance(s1)
print(f'Standard Deviation of sample is {stddev}')
print(f'Variance of sample is {var}')
```

```
Standard Deviation of sample is 318.30586268905296
Variance of sample is 101318.62222222223
```

In [ ]:

```
# Q. You have 8 numbers. The mean is 6. You add 5 to each number in the group. New Mean=?
## A. New Mean=old mean+5=11
```

In [ ]:

```
# Q. You have 15 numbers. The mean is 10, and the variance is 4. You multiply each number
by 3. What is the new standard
# deviation?
## A. The new standard deviation will be: 3 x sqrt(4) = 6aYou have 15 numbers. The mean i
s 10, and the variance is 4.
```

In [9]:

```
# #Temperature of 5 cities are given, from the given values,what would be the mean and st
andard deviation of temperature
# in Celsius?
temp=[82,77,41,78,84]
mean=np.mean(temp)
stdev=statistics.stdev(temp)
print('In Fahrenheit')
print(f'Mean : {mean} F')
print(f'Standard deviation : {stdev} F')
print('\nIn Celcius')
print(f'Mean : {mean*0.556 - 17.778} C')
print(f'Standard deviation : {stdev * 0.556} C')
```

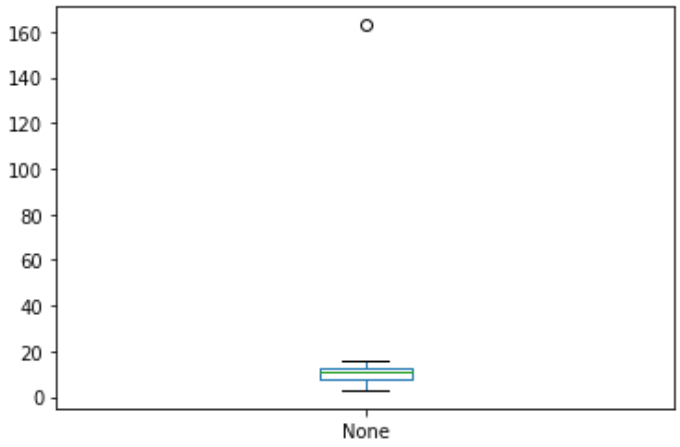
```
In Fahrenheit
Mean : 72.4 F
```

```
Standard deviation : 17.78482499211055 F

In Celcius
Mean : 22.476400000000005 C
Standard deviation : 9.888362695613466 C
```

In [10]:

```
# Q.6 Construct a boxplot for the following data set.
# 3, 5, 8, 8, 9, 11, 12, 12, 13, 13, 163,5,8,8,9,11,12,12,13,13,16

data=[3, 5, 8, 8, 9, 11, 12, 12, 13, 13, 163,5,8,8,9,11,12,12,13,13,16]
df=pd.Series(data)
print(df.plot.box())
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [32]:

```
# Consider below dataset, calculate the skewness and then tell if it is left skewed or ri
ght skewed?
# 12, 13, 54, 56 , 25
dataset=[12, 13, 54, 56 , 25]
print(f'Skewness:{skew(dataset)}')
mean1=np.mean(dataset)
median1=np.median(dataset)
if(median1<mean1):
                        print("Right Skewed")
else:
                        print("Left Skewed")
```

```
Skewness:0.2563317051472635
Right Skewed
```

In [39]:

```
#Determine the excess-kurtosis of the below dataset and then categorize it according to t
he type of kurtosis it is and give its characteristics.
# 12, 13, 54, 56 , 25
data=[12, 13, 54, 56 , 25]
kurtosis1={kurtosis(data)}
print(kurtosis1)
print("Type: platykurtic")
```

```
{-1.7721210214761647}
Type: platykurtic
```

In [41]:

```
# Determine the outliers in the below dataset using IQR formula.
# 1, 99, 100, 101, 103, 109, 110, 201
data=[1, 99, 100, 101, 103, 109, 110, 201]
Q1,Q3=np.percentile(data,[25,75],interpolation='midpoint')
print(f'1st Quartile is {Q1}')
print(f'3rd Quartile is {Q3}')
```

```
IQR=Q3-Q1
print(f'IQR is {IQR}')
outlier=[i for i in data if i<Q1-1.5*IQR or i>Q3+1.5*IQR]
print(f'\nOutliers: {outlier}')
```

```
1st Quartile is 99.5
3rd Quartile is 109.5
IQR is 10.0

Outliers: [1, 201]
```

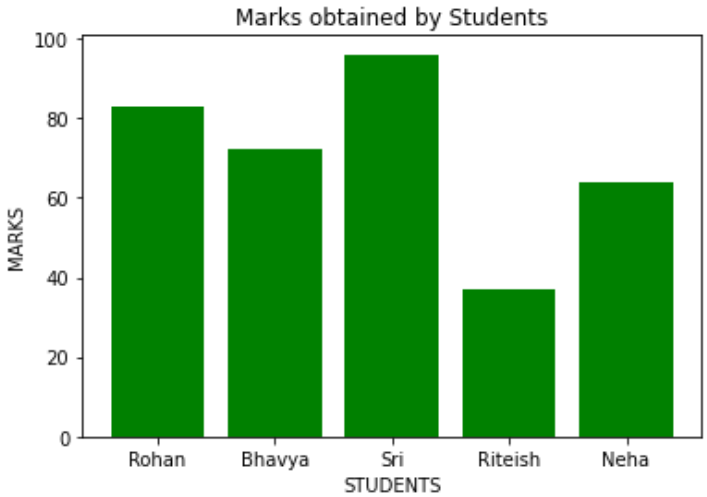
In [45]:

```
# Determine the outlier in the below dataset using Z-score.
# 1.5895, 1.6508, 1.7131, 1.7136,1.7212, 1.7296, 1.7343, 1.7663, 1.8018, 1.8394,
# 1.8869, 1.9357, 1.9482, 2.1038, 10.8135, -0.0012
data=[1.5895, 1.6508, 1.7131, 1.7136,1.7212, 1.7296, 1.7343, 1.7663, 1.8018, 1.8394, 1.8
869, 1.9357, 1.9482, 2.1038, 10.8135, -0.0012]
outlier=[]
mean=np.mean(data)
stdev=statistics.stdev(data)
print(f'Mean of dataset is {mean}')
print(f'Standard deviation of dataset is {stdev}')
def zscore(x):
    z=(x-mean)/stdev
    return z
for i in data:
    if zscore(i)>2 or zscore(i)<-2:
        outlier.append(i)
    else:
        continue
print(f'\nOutliers is {outlier}')
```

```
Mean of dataset is 2.24665625
Standard deviation of dataset is 2.331458190623413

Outliers is [10.8135]
```

In [50]:

```
# Below is the mark obtained by some students. Construct a bar chart for it:
# Student Marks
# Rohan 83
# Bhavya 72
# Sri 96
# Riteish 37
# Neha 64
plt.bar(['Rohan','Bhavya','Sri','Riteish','Neha'],[83,72,96,37,64],color='g')
plt.xlabel('STUDENTS',color='black')
plt.ylabel('MARKS',color='black')
plt.title('Marks obtained by Students')
plt.show()
```



In [55]:

```
# Below are some observations obtained from a hospital and shows glucose level o
# f some patients.
# Check the correlation of the variables and then tell if it has positive, negative or no
# relation between them.
# Calculate co-variance and also plot a scatter-plot to see the relation visually.
# Patient Age (years) Glucose Level
# A 46 99
# B 24 65
# C 28 78
# D 42 79
# E 59 89
# F 48 82
patient=['A','B','C','D','E','F']
age=[46,24,28,42,59,48]
Glucose_level=[99,65,78,79,89,82]
Detail=list(zip(age,Glucose_level))
df=pd.DataFrame(Detail)
Correlation=df.corr(method='pearson')
print(f'Correlation:\n{Correlation}')
print('\nPositive correlation.\n')
covariance=np.cov(age,Glucose_level)
print(f'Covariance matrix :\n{covariance}')
plt.scatter(age,Glucose_level,color='g')
plt.xlabel('Age')
plt.ylabel('Glucose Level')
plt.title('Glucose Level Vs Age')
plt.show()
```

```
Correlation:
          0         1
0  1.000000  0.734514
1  0.734514  1.000000

Positive correlation.

Covariance matrix :
[[171.36666667 109.8       ]
 [109.8        130.4       ]]
```

In [56]:

```
# Try to findout if there is any correlation between Physical Activity and Blood Pressure
. Calculate Spearman Rank Correlation.

Pa=[60,55,25,50,40,45,35,10,30,20]
BP=[118,117,120,121,119,122,123,124,125,126]
Detail=list(zip(Pa,BP))
df=pd.DataFrame(Detail)
cor=df.corr(method='spearman')
print(cor)
```

```
          0         1
0  1.000000 -0.757576
1 -0.757576  1.000000
```