In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in ./lib/python3.8/site-packages (1.22.0)
```

In [5]:
```
df = pd.read_csv (r'/home/vedant/Downloads/StudentsPerformance_modified.csv')
print(df)
```

```
     gender race/ethnicity parental level of education         lunch  \
0    female        group B           bachelor's degree      standard
1    female        group C                some college      standard
2    female        group B            master's degree      standard
3      male        group A         associate's degree  free/reduced
4      male        group C                some college      standard
..      ...            ...                         ...           ...
995  female        group E            master's degree      standard
996    male        group C                high school  free/reduced
997  female        group C                high school  free/reduced
998  female        group D               some college      standard
999  female        group D               some college  free/reduced

    test preparation course  math_score  reading score  writing score
0                      none          72           72.0           74.0
1                 completed          69          900.0          880.0
2                      none          90          950.0           93.0
3                      none          47           57.0           44.0
4                      none          76           78.0           75.0
..                      ...         ...            ...            ...
995               completed          88           99.0           95.0
996                    none          62           55.0           55.0
997               completed          59           71.0           65.0
998               completed          68           78.0           77.0
999                    none          77           86.0           86.0

[1000 rows x 8 columns]
```

In [6]:
```
readingmean = np.mean(df["reading score"])
print(readingmean)
```

```
72.09657947686117
```

In [7]:
```
readingstd = np.std(df['reading score'])
print(readingstd)
```

```
60.685824905300926
```

In [8]:
```
col_list = ["reading score"]
readingdata = pd.read_csv (r'/home/vedant/Downloads/StudentsPerformance_modi1
print(readingdata)
```

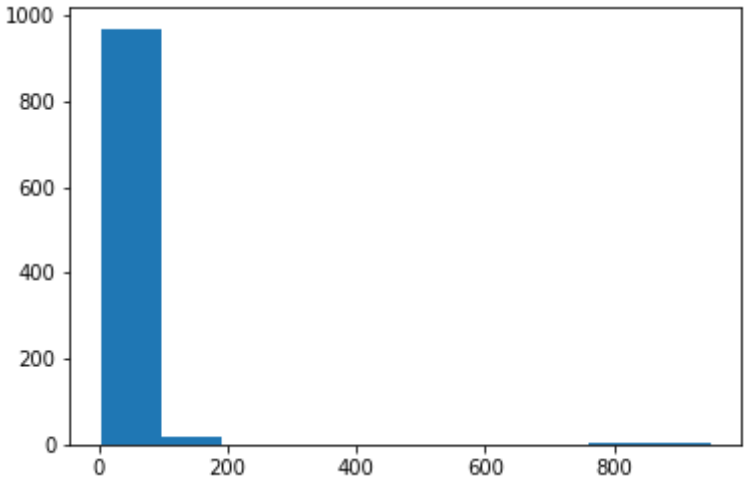
```
     reading score
0             72.0
1            900.0
```

```
2            950.0
3             57.0
4             78.0
..             ...
995           99.0
996           55.0
997           71.0
998           78.0
999           86.0

[1000 rows x 1 columns]
```

In [10]:

```
plt.hist(readingdata)
plt.show()
```



DETECTING OUTLIERS USING Z SCORE TECHNIQUE

Z score is an important concept in statistics. Z score is also called standard score. This score helps to understand if a data value is greater or smaller than mean and how far away it is from the mean. More specifically, Z score tells how many standard deviations away a data point is from the mean.

Z score = (x -mean) / std. deviation

In [11]:

```
outlier = []
def findoutliers(readingdata):

    readingmean = np.mean(df["reading score"])
    readingstd = np.std(df['reading score'])
    threshold = 3

    for i in readingdata["reading score"]:

        z = (i-readingmean)/readingstd

        if z > threshold:
            outlier.append(i)
```

In [12]:

```
findoutliers(readingdata)
print(outlier)
```

```
[900.0, 950.0, 830.0, 930.0, 850.0]
```

----------using lower and upperbounds----------

In [18]:
```
lb=0.4
ub=0.8
res = readingdata.quantile([lb,ub])
res
```

Out[18]:

	reading score
0.4	65.0
0.8	82.0

In [31]:
```
type(res)
```

Out[31]:
```
pandas.core.frame.DataFrame
```

In [34]:
```
valueAtLb = res['reading score'].iloc[0]
valueAtLb
```

Out[34]:
```
65.0
```

In [35]:
```
valueAtUb = res['reading score'].iloc[1]
valueAtUb
```

Out[35]:
```
82.0
```

In [36]:
```
true_index=(valueAtLb<df["reading score"].values) & \
           (df["reading score"].values<valueAtUb)
true_index
```

Out[36]:
```
array([ True, False, False, False,  True, False, False, False, False,
       False, False, False,  True,  True, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False,  True, False, False, False,
       False, False,  True,  True, False, False,  True, False,  True,
       False, False, False, False, False, False,  True, False, False,
        True, False, False, False,  True, False, False, False, False,
       False, False, False, False, False,  True,  True,  True, False,
       False, False, False, False,  True, False,  True,  True, False,
        True, False,  True, False, False,  True,  True,  True,  True,
        True,  True,  True, False, False, False,  True, False, False,
        True, False, False, False, False, False, False,  True, False,
       False, False,  True, False, False, False, False,  True, False,
        True,  True, False, False, False, False,  True,  True,  True,
       False, False, False,  True, False,  True, False, False, False,
        True, False, False,  True,  True, False,  True,  True,  True,
       False,  True, False,  True, False, False, False,  True, False,
       False, False, False, False, False,  True, False,  True,  True,
       False, False,  True, False, False, False,  True,  True, False,
        True, False, False,  True, False,  True,  True,  True, False,
       False, False, False, False, False, False, False,  True, False,
       False,  True, False,  True,  True,  True, False,  True, False,
        True,  True, False,  True, False, False, False, False, False,
       False, False,  True, False,  True, False, False,  True,  True,
       False,  True, False,  True, False, False, False, False, False,
       False,  True,  True, False, False,  True,  True, False, False,
       False,  True,  True,  True,  True, False, False, False, False,
```

```
 True,  True, False,  True,  True,  True,  True,  True,  True,
 True, False, False,  True, False,  True,  True, False,  True,
False, False, False,  True, False,  True, False, False, False,
False, False, False,  True,  True, False, False, False, False,
 True,  True,  True,  True, False,  True,  True, False, False,
 True, False, False,  True, False, False,  True,  True,  True,
False, False,  True, False,  True, False, False,  True, False,
 True, False,  True,  True, False,  True, False,  True, False,
False, False, False, False,  True, False, False, False, False,
 True, False,  True,  True, False, False, False, False,  True,
 True,  True,  True,  True, False, False,  True,  True, False,
False,  True, False,  True,  True, False,  True, False, False,
False, False, False, False, False, False, False,  True, False,
False,  True,  True,  True, False,  True, False, False, False,
False, False, False, False,  True, False, False,  True,  True,
 True, False,  True,  True,  True,  True, False,  True, False,
 True, False, False, False, False, False, False, False, False,
 True, False,  True, False, False,  True, False,  True,  True,
 True,  True,  True,  True, False, False, False, False, False,
False, False,  True, False,  True, False, False,  True,  True,
False, False,  True, False,  True,  True, False, False,  True,
 True,  True, False, False,  True,  True,  True, False,  True,
False, False,  True, False, False, False, False, False, False,
False, False, False,  True, False, False, False, False,  True,
False,  True, False,  True, False,  True, False,  True, False,
 True, False,  True,  True, False,  True, False, False,  True,
False,  True, False, False, False, False, False, False, False,
False, False,  True,  True,  True,  True, False, False, False,
False, False,  True, False,  True, False,  True, False, False,
False, False, False, False,  True,  True,  True, False, False,
False, False, False, False, False, False, False,  True,  True,
False, False, False,  True, False, False,  True,  True, False,
 True, False,  True, False, False, False, False,  True,  True,
 True,  True, False, False, False, False, False,  True,  True,
 True,  True,  True,  True, False,  True, False, False, False,
 True, False,  True,  True, False, False,  True,  True, False,
False, False, False, False, False,  True,  True,  True,  True,
 True,  True, False, False,  True, False, False,  True,  True,
False,  True, False, False,  True, False, False, False,  True,
False, False,  True, False, False, False, False, False, False,
False,  True, False,  True, False,  True,  True,  True, False,
False, False, False, False, False,  True, False, False, False,
False, False,  True, False, False,  True,  True, False,  True,
 True, False, False,  True,  True, False,  True,  True, False,
False,  True, False,  True, False,  True,  True,  True,  True,
 True, False, False,  True,  True,  True,  True, False, False,
 True, False,  True,  True,  True, False,  True, False,  True,
False,  True, False,  True, False,  True,  True, False, False,
 True, False,  True,  True, False, False, False, False, False,
False, False, False, False,  True,  True,  True,  True,  True,
False, False, False,  True, False, False, False, False, False,
False, False,  True,  True, False,  True, False,  True,  True,
 True, False, False, False, False,  True,  True, False, False,
False,  True, False, False, False, False, False,  True,  True,
 True, False,  True, False, False,  True, False,  True,  True,
False, False, False,  True,  True,  True, False, False, False,
False,  True, False,  True,  True, False,  True, False, False,
 True,  True, False,  True, False, False,  True,  True,  True,
False, False,  True, False,  True, False, False, False, False,
False,  True, False,  True, False, False, False, False,  True,
 True, False, False,  True,  True, False, False, False,  True,
 True, False, False,  True,  True,  True, False,  True, False,
False, False, False, False,  True, False, False,  True,  True,
False, False, False, False, False, False,  True,  True,  True,
```

```
       False, False, False, False, False, False, False, False, False,
       False, False,  True, False, False, False,  True, False, False,
       False, False,  True,  True,  True,  True, False,  True, False,
       False,  True,  True, False,  True, False, False, False,  True,
       False, False, False, False,  True, False, False,  True, False,
       False, False, False, False, False,  True,  True, False,  True,
        True, False, False,  True, False,  True, False, False, False,
       False,  True,  True, False, False, False,  True, False,  True,
       False,  True, False, False, False,  True, False, False,  True,
       False, False, False, False, False, False,  True, False, False,
        True, False,  True, False,  True, False,  True, False, False,
       False, False, False,  True, False,  True,  True, False, False,
       False, False,  True,  True,  True, False,  True, False,  True,
       False, False, False, False,  True,  True,  True,  True, False,
        True, False, False, False, False,  True, False, False, False,
        True, False,  True,  True, False,  True, False, False,  True,
       False, False, False, False, False, False, False, False, False,
        True, False, False,  True, False, False,  True, False, False,
        True, False,  True,  True, False, False, False,  True,  True,
       False])
```

In [37]:

```
false_index=~true_index
false_index
```

Out[37]:

```
array([False,  True,  True,  True, False,  True,  True,  True,  True,
        True,  True,  True, False, False,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True, False,  True,  True,  True,
        True,  True, False, False,  True,  True, False,  True, False,
        True,  True,  True,  True,  True,  True, False,  True,  True,
       False,  True,  True,  True, False,  True,  True,  True,  True,
        True,  True,  True,  True,  True, False, False, False,  True,
        True,  True,  True,  True, False,  True, False, False,  True,
       False,  True, False,  True,  True, False, False, False, False,
       False, False, False,  True,  True,  True, False,  True,  True,
       False,  True,  True,  True,  True,  True,  True, False,  True,
        True,  True, False,  True,  True,  True,  True, False,  True,
       False, False,  True,  True,  True,  True, False, False, False,
        True,  True,  True, False,  True, False,  True,  True,  True,
       False,  True,  True, False, False,  True, False, False, False,
        True, False,  True, False,  True,  True,  True, False,  True,
        True,  True,  True,  True,  True, False,  True, False, False,
        True,  True, False,  True,  True,  True, False, False,  True,
       False,  True,  True, False,  True, False, False, False,  True,
        True,  True,  True,  True,  True, False,  True, False,  True,
       False, False,  True, False,  True,  True,  True,  True,  True,
        True,  True, False,  True, False,  True,  True, False, False,
        True, False,  True, False,  True,  True,  True,  True,  True,
        True, False, False,  True,  True, False, False,  True,  True,
        True, False, False, False, False,  True,  True,  True,  True,
       False, False,  True, False, False, False, False, False, False,
       False,  True,  True, False,  True, False, False,  True, False,
        True,  True,  True, False,  True, False,  True,  True,  True,
        True,  True,  True, False, False,  True,  True,  True,  True,
       False, False, False, False,  True, False, False,  True,  True,
       False,  True,  True, False,  True,  True, False, False, False,
        True,  True, False,  True, False,  True,  True, False,  True,
       False,  True, False, False,  True, False,  True, False,  True,
        True,  True,  True,  True, False,  True,  True,  True,  True,
       False,  True, False, False,  True,  True,  True,  True, False,
       False, False, False, False,  True,  True, False, False,  True,
```

```
 True, False,  True, False, False,  True, False,  True,  True,
 True,  True,  True,  True,  True,  True,  True, False,  True,
 True, False, False, False,  True, False,  True,  True,  True,
 True,  True,  True,  True, False,  True,  True, False, False,
False,  True, False, False, False, False,  True, False,  True,
False,  True,  True,  True,  True,  True,  True,  True,  True,
False,  True, False,  True,  True, False,  True, False, False,
False, False, False, False,  True,  True,  True,  True,  True,
 True,  True, False,  True, False,  True,  True, False, False,
 True,  True, False,  True, False, False,  True,  True, False,
False, False,  True,  True, False, False, False,  True, False,
 True,  True, False,  True,  True,  True,  True,  True,  True,
 True,  True,  True, False,  True,  True,  True,  True, False,
 True, False,  True, False,  True, False,  True, False,  True,
False,  True, False, False,  True, False,  True,  True, False,
 True, False,  True,  True,  True,  True,  True,  True,  True,
 True,  True, False, False, False, False,  True,  True,  True,
 True,  True, False,  True, False,  True, False,  True,  True,
 True,  True,  True,  True, False, False, False,  True,  True,
 True,  True,  True,  True,  True,  True,  True, False, False,
 True,  True,  True, False,  True,  True, False, False,  True,
False,  True, False,  True,  True,  True,  True, False, False,
False, False,  True,  True,  True,  True,  True, False, False,
False, False, False, False,  True, False,  True,  True,  True,
False,  True, False, False,  True,  True, False, False,  True,
 True,  True,  True,  True,  True, False, False, False, False,
False, False,  True,  True, False,  True,  True, False, False,
 True, False,  True,  True, False,  True,  True,  True, False,
 True,  True, False,  True,  True,  True,  True,  True,  True,
 True, False,  True, False,  True, False, False, False,  True,
 True,  True,  True,  True,  True, False,  True,  True,  True,
 True,  True, False,  True,  True, False, False,  True, False,
False,  True, False, False, False,  True, False,  True, False,
 True, False,  True, False,  True, False, False,  True,  True,
False,  True, False, False, False,  True,  True,  True,  True,
 True,  True,  True,  True, False, False, False, False, False,
 True,  True,  True, False,  True,  True,  True,  True,  True,
 True,  True, False, False,  True, False,  True, False, False,
False,  True,  True,  True,  True, False, False,  True,  True,
 True, False,  True,  True,  True,  True,  True, False, False,
False,  True, False,  True,  True, False,  True, False, False,
 True,  True,  True, False, False, False,  True,  True,  True,
 True, False,  True, False, False,  True, False,  True,  True,
False, False,  True, False,  True,  True, False, False, False,
 True,  True, False,  True, False,  True,  True,  True,  True,
 True, False,  True, False,  True,  True,  True,  True, False,
False,  True,  True, False, False,  True,  True,  True, False,
False,  True,  True, False, False, False,  True, False,  True,
 True,  True,  True,  True, False,  True,  True, False, False,
 True,  True,  True,  True,  True,  True, False, False, False,
 True,  True,  True,  True,  True,  True,  True,  True,  True,
 True,  True, False,  True,  True,  True, False,  True,  True,
 True,  True, False, False, False, False,  True, False,  True,
 True, False, False,  True, False,  True,  True,  True, False,
 True,  True,  True,  True, False,  True,  True, False,  True,
 True,  True,  True,  True,  True, False, False,  True, False,
False,  True,  True, False,  True, False,  True,  True,  True,
 True, False, False,  True,  True,  True, False,  True, False,
 True, False,  True,  True,  True, False,  True,  True, False,
 True,  True,  True,  True,  True,  True, False,  True,  True,
False,  True, False,  True, False,  True, False,  True,  True,
```

```
 True,  True,  True, False,  True, False, False,  True,  True,
 True,  True, False, False, False,  True, False,  True, False,
 True,  True,  True,  True, False, False, False, False,  True,
False,  True,  True,  True,  True, False,  True,  True,  True,
False,  True, False, False,  True, False,  True,  True, False,
 True,  True,  True,  True,  True,  True,  True,  True,  True,
False,  True,  True, False,  True,  True, False,  True,  True,
False,  True, False, False,  True,  True,  True, False, False,
 True])
```

In [39]:
```
readingdata[true_index]
```

Out[39]:

	reading score
0	72.0
4	78.0
12	81.0
13	72.0
41	73.0
...	...
990	81.0
992	76.0
993	72.0
997	71.0
998	78.0

382 rows × 1 columns

In [40]:
```
mymedian = np.median(readingdata[true_index])
mymedian
```

Out[40]:
```
73.0
```

In [41]:
```
readingdata[false_index]  = mymedian
```

after successfully removing outliers (replacing outliers with median)

In [42]:
```
readingdata
```

Out[42]:

	reading score
0	72.0
1	73.0
2	73.0
3	73.0
4	78.0
...	...
995	73.0

	reading score
996	73.0
997	71.0
998	78.0
999	73.0

1000 rows × 1 columns

In [ ]: