In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:

```
df= pd.read_csv("student_scores.csv")
print(df.head)
```

```
<bound method NDFrame.head of        Unnamed: 0  Gender EthnicGroup
ParentEduc     LunchType  \
0                0  female         NaN   bachelor's degree      standard
1                1  female     group C        some college      standard
2                2  female     group B     master's degree      standard
3                3    male     group A  associate's degree  free/reduced
4                4    male     group C        some college      standard
...            ...     ...         ...                 ...           ...
30636          816  female     group D         high school      standard
30637          890    male     group E         high school      standard
30638          911  female         NaN         high school  free/reduced
30639          934  female     group D  associate's degree      standard
30640          960    male     group B        some college      standard

        TestPrep ParentMaritalStatus PracticeSport IsFirstChild  NrSiblings
\
0           none             married     regularly          yes         3.0
1            NaN             married     sometimes          yes         0.0
2           none              single     sometimes          yes         4.0
3           none             married         never           no         1.0
4           none             married     sometimes          yes         0.0
...          ...                 ...           ...          ...         ...
30636       none              single     sometimes           no         2.0
30637       none              single     regularly           no         1.0
30638  completed             married     sometimes           no         1.0
30639  completed             married     regularly           no         3.0
30640       none             married         never           no         1.0

      TransportMeans WklyStudyHours  MathScore  ReadingScore  WritingScore
0         school_bus            < 5         71            71            74
1                NaN         5 - 10         69            90            88
2         school_bus            < 5         87            93            91
3                NaN         5 - 10         45            56            42
4         school_bus         5 - 10         76            78            75
...              ...            ...        ...           ...           ...
30636     school_bus         5 - 10         59            61            65
30637        private         5 - 10         58            53            51
30638        private         5 - 10         61            70            67
30639     school_bus         5 - 10         82            90            93
30640     school_bus         5 - 10         64            60            58

[30641 rows x 15 columns]>
```

In [12]:

```
df.describe()
```

Out[12]:

	Unnamed: 0	NrSiblings	MathScore	ReadingScore	WritingScore
count	30641.000000	29069.000000	30641.000000	30641.000000	30641.000000
mean	499.556607	2.145894	66.558402	69.377533	68.418622
std	288.747894	1.458242	15.361616	14.758952	15.443525
min	0.000000	0.000000	0.000000	10.000000	4.000000
25%	249.000000	1.000000	56.000000	59.000000	58.000000
50%	500.000000	2.000000	67.000000	70.000000	69.000000
75%	750.000000	3.000000	78.000000	80.000000	79.000000
max	999.000000	7.000000	100.000000	100.000000	100.000000

In [13]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Unnamed: 0           30641 non-null  int64  
 1   Gender               30641 non-null  object 
 2   EthnicGroup          28801 non-null  object 
 3   ParentEduc           28796 non-null  object 
 4   LunchType            30641 non-null  object 
 5   TestPrep             28811 non-null  object 
 6   ParentMaritalStatus  29451 non-null  object 
 7   PracticeSport        30010 non-null  object 
 8   IsFirstChild         29737 non-null  object 
 9   NrSiblings           29069 non-null  float64
 10  TransportMeans       27507 non-null  object 
 11  WklyStudyHours       29686 non-null  object 
 12  MathScore            30641 non-null  int64  
 13  ReadingScore         30641 non-null  int64  
 14  WritingScore         30641 non-null  int64  
dtypes: float64(1), int64(4), object(10)
memory usage: 3.5+ MB
```

In [16]:

```
df.isnull().sum() #gives count of null values
```

Out[16]:

```
Unnamed: 0                0
Gender                    0
EthnicGroup            1840
ParentEduc             1845
LunchType                 0
TestPrep               1830
ParentMaritalStatus    1190
PracticeSport           631
IsFirstChild            904
NrSiblings             1572
TransportMeans         3134
WklyStudyHours          955
MathScore                 0
ReadingScore              0
WritingScore              0
dtype: int64
```

DROP UNNAMED COLUMN

In [17]:

```
df = df.drop("Unnamed: 0" , axis = 1)
print (df.head())
```

```
   Gender EthnicGroup          ParentEduc     LunchType TestPrep  \
0  female         NaN   bachelor's degree      standard     none   
1  female     group C        some college      standard      NaN   
2  female     group B     master's degree      standard     none   
3    male     group A  associate's degree  free/reduced     none   
4    male     group C        some college      standard     none   

  ParentMaritalStatus PracticeSport IsFirstChild  NrSiblings TransportMeans  
\
0             married     regularly          yes         3.0     school_bus  
1             married     sometimes          yes         0.0            NaN  
2              single     sometimes          yes         4.0     school_bus  
3             married         never           no         1.0            NaN  
4             married     sometimes          yes         0.0     school_bus  

  WklyStudyHours  MathScore  ReadingScore  WritingScore  
0            < 5         71            71            74  
1         5 - 10         69            90            88  
2            < 5         87            93            91  
3         5 - 10         45            56            42  
4         5 - 10         76            78            75  
```

CHANGE WEEKLY STUDY HOURS

In [22]:

```
df["WklyStudyHours"] = df["WklyStudyHours"].str.replace("5 - 10" , "6-10")
df.head()
```

Out[22]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	Pra
0	0	female	NaN	bachelor's degree	standard	none	married	
1	1	female	group C	some college	standard	NaN	married	
2	2	female	group B	master's degree	standard	none	single	
3	3	male	group A	associate's degree	free/reduced	none	married	
4	4	male	group C	some college	standard	none	married	

In [25]:

```
df["WklyStudyHours"] = df["WklyStudyHours"].str.replace("6-10" , "5-10")
df.head()
```

Out[25]:

amed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSpor
0	female	NaN	bachelor's degree	standard	none	married	regularl
1	female	group C	some college	standard	NaN	married	sometime
2	female	group B	master's degree	standard	none	single	sometime
3	male	group A	associate's degree	free/reduced	none	married	neve
4	male	group C	some college	standard	none	married	sometime

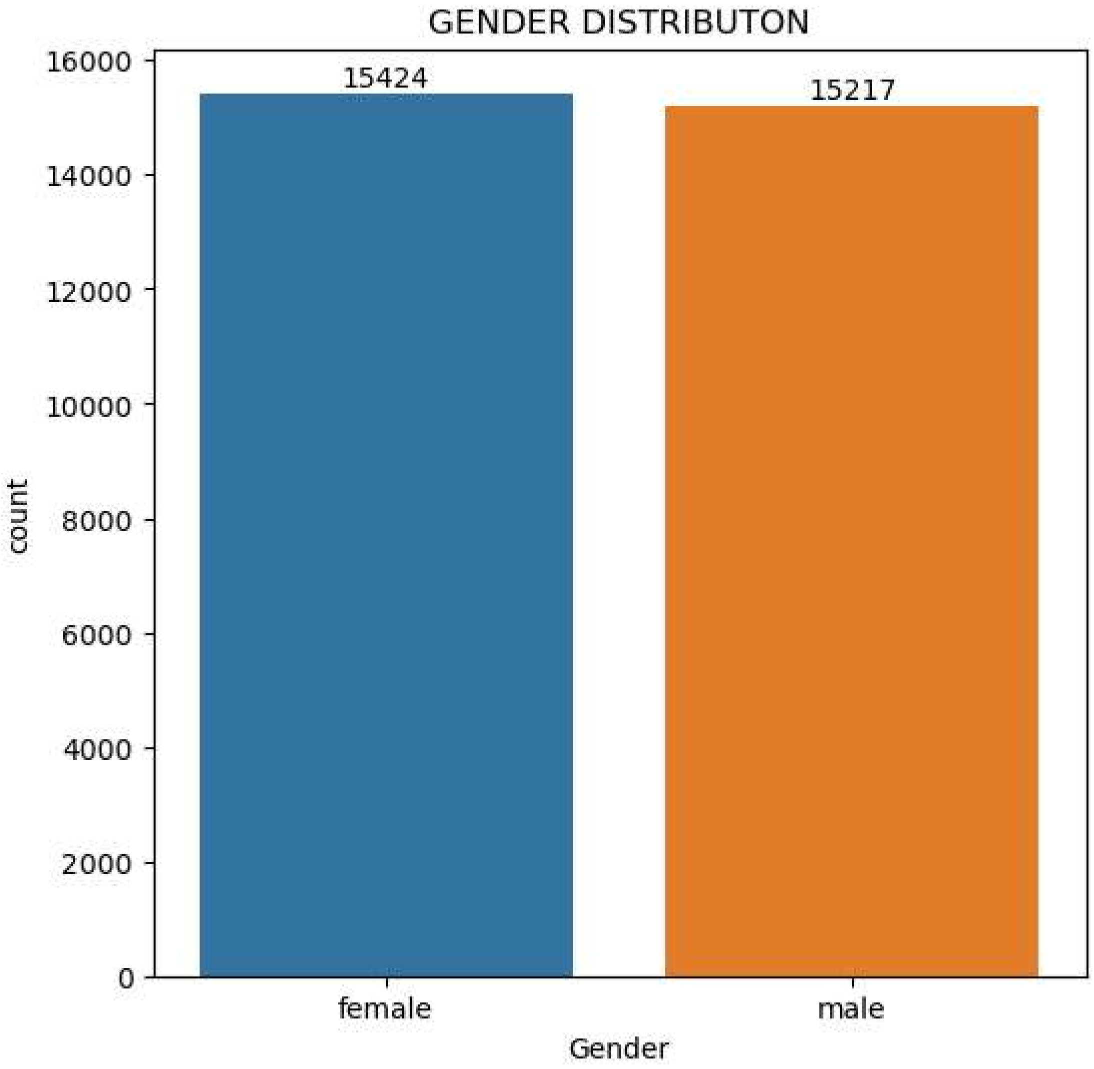
Gender distribution

In [50]:

```
plt.figure(figsize= (6,6))
ax = sns.countplot(data= df, x= "Gender")
plt.title ("GENDER DISTRIBUTON")
ax.bar_label(ax.containers[0]) #show the number of count
plt.show
```

Out[50]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



FROM THE ABOVE CHART WE CAN SAY THAT:

THE NUMBER OF FEMALES IS MORE THAN MALES

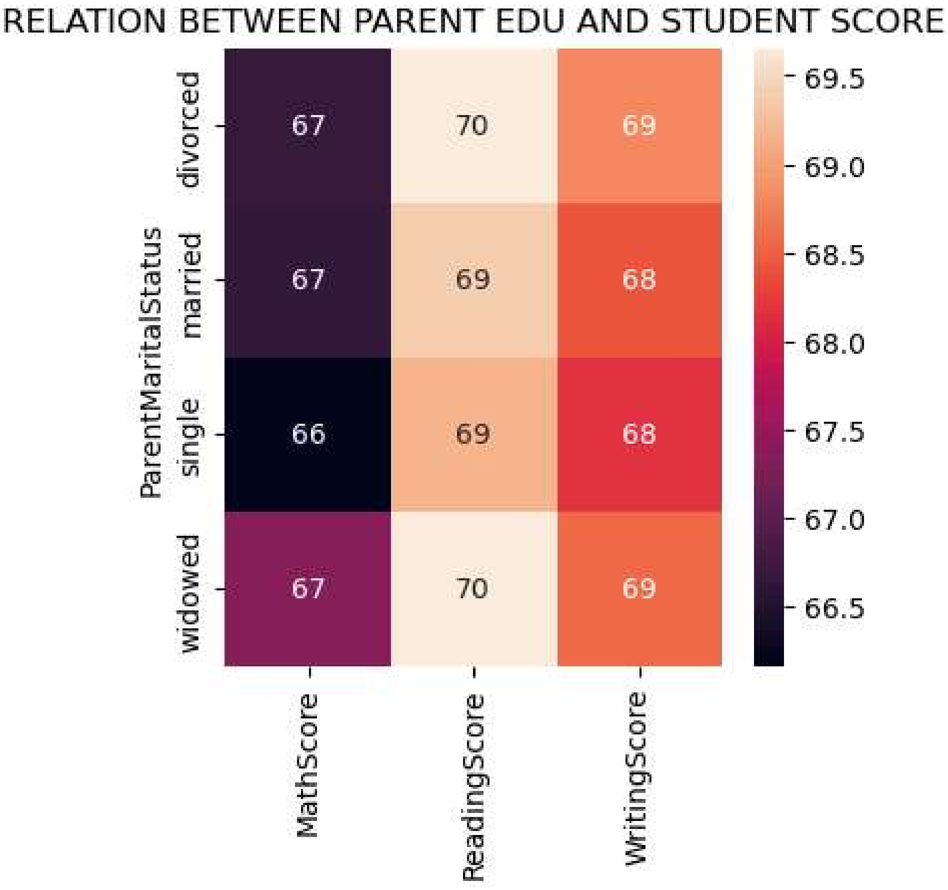
In [36]:

```
gb = df.groupby ("ParentEduc").agg({"MathScore":"mean" , "ReadingScore" :"
print(gb)
```

```
                    MathScore  ReadingScore  WritingScore
ParentEduc
associate's degree  68.365586     71.124324     70.299099
bachelor's degree   70.466627     73.062020     73.331069
high school         64.435731     67.213997     65.421136
master's degree     72.336134     75.832921     76.356896
some college        66.390472     69.179708     68.501432
some high school    62.584013     65.510785     63.632409
```

In [51]:

```
plt.figure (figsize= (4,4))
plt.title ("RELATION BETWEEN PARENT EDU AND STUDENT SCORE")
sns.heatmap(gb, annot= True) #shows value of cells
plt.show()
```



RELATION BETWEEN PARENT EDU AND STUDENT SCORE

In [ ]:

```
FROM THE ABOVE CHART WE HAVE CONCLUDED THAT
THE PARENTS EDUCATION HAS A HIGH IMPACT ON THE STUDENT SCORE
```

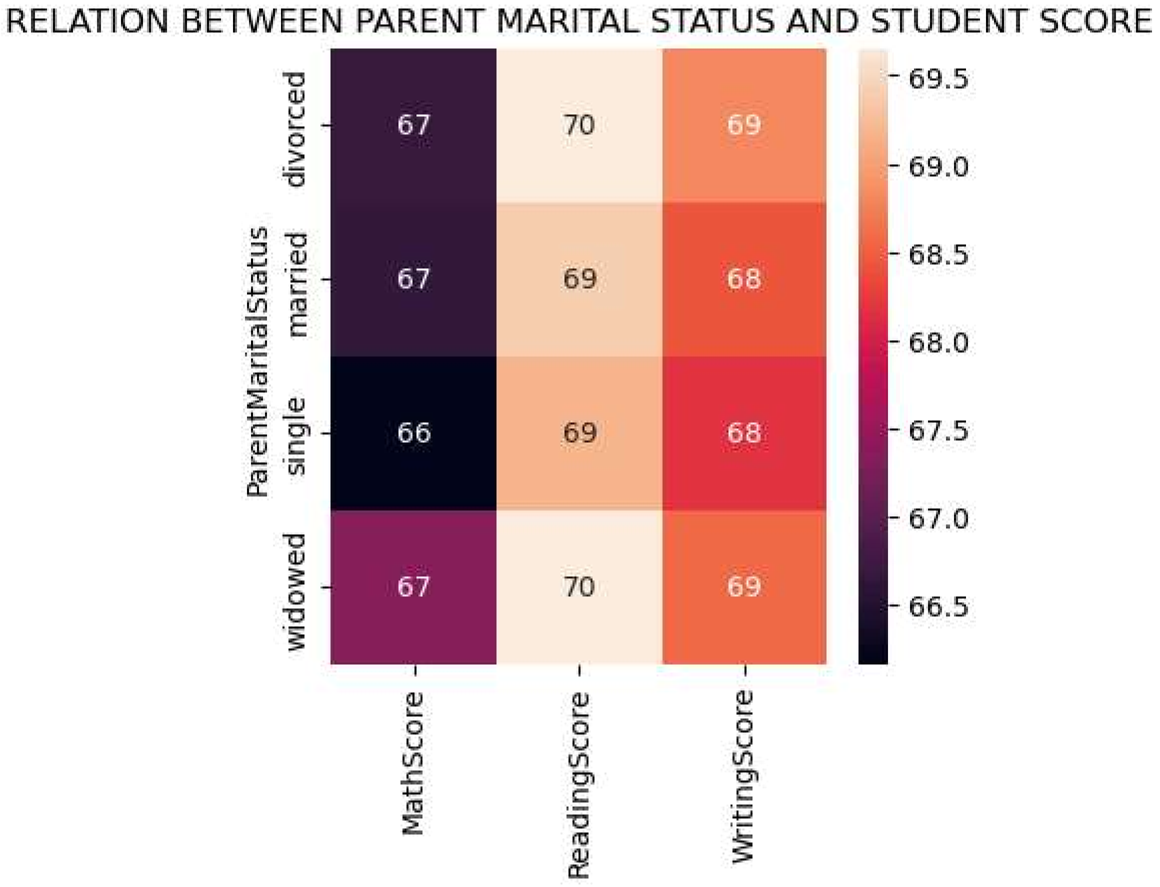
In [44]:

```
gb1= df.groupby ("ParentMaritalStatus").agg({"MathScore":"mean" , "Readin
print(gb1)
```

```
                     MathScore  ReadingScore  WritingScore
ParentMaritalStatus
divorced             66.691197     69.655011     68.799146
married              66.657326     69.389575     68.420981
single               66.165704     69.157250     68.174440
widowed              67.368866     69.651438     68.563452
```

In [52]:

```
plt.figure (figsize= (4,4))
sns.heatmap(gb1, annot= True) #shows value of cells
plt.title ("RELATION BETWEEN PARENT MARITAL STATUS AND STUDENT SCORE")
plt.show()
```


RELATION BETWEEN PARENT MARITAL STATUS AND STUDENT SCORE

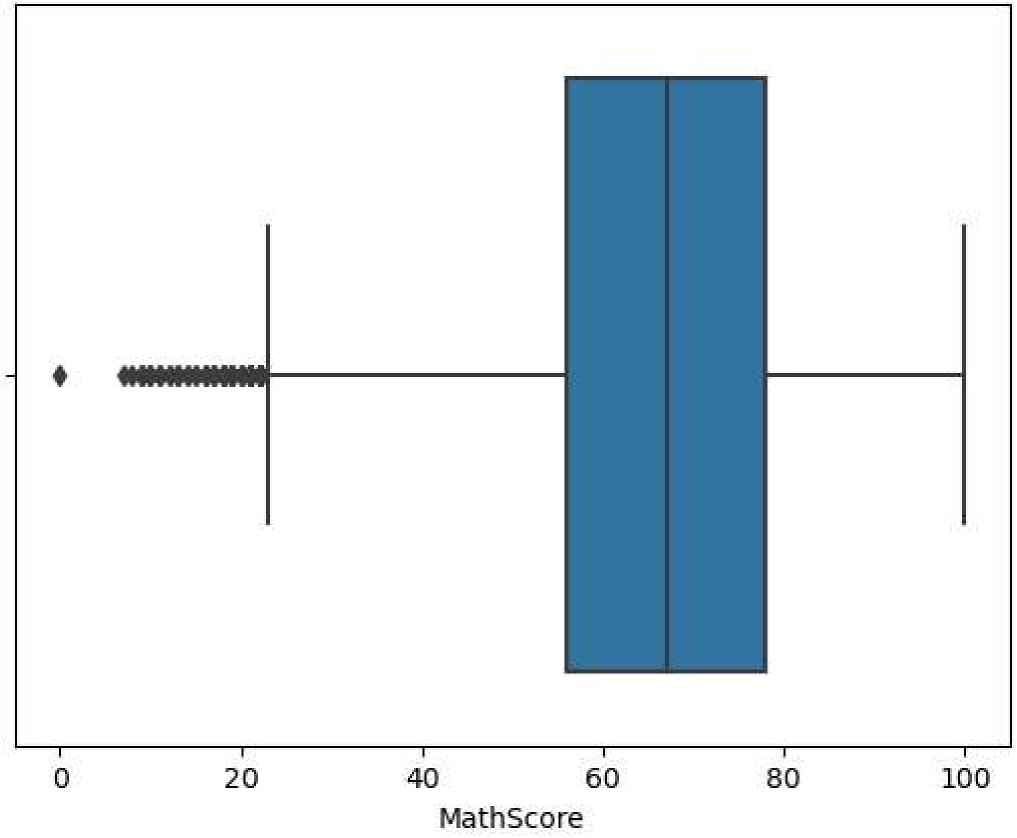
In [ ]:

```
WE CAN STATE THAT
THE MARITAL STATUS HAVE NEGLIGICLE EFFECT ON THE STUDENT SCORE
```

TO DETECT OUTLINERS

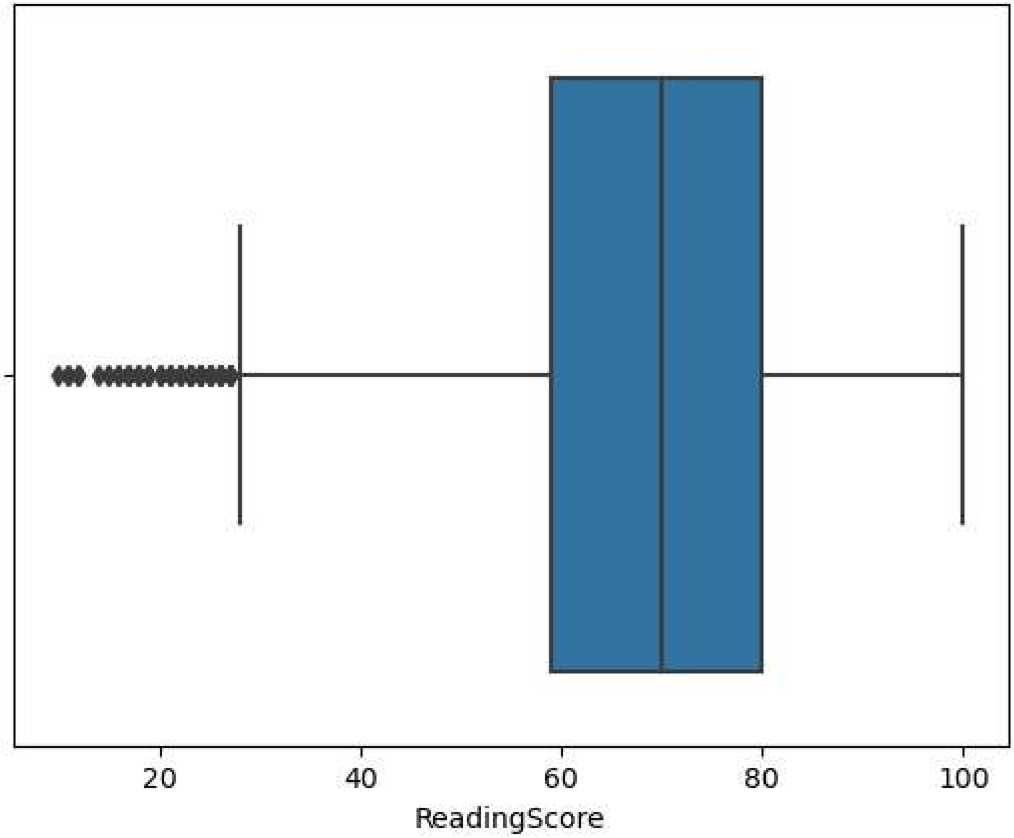
In [54]:

```
sns.boxplot (data = df , x = "MathScore")
plt.show()
```



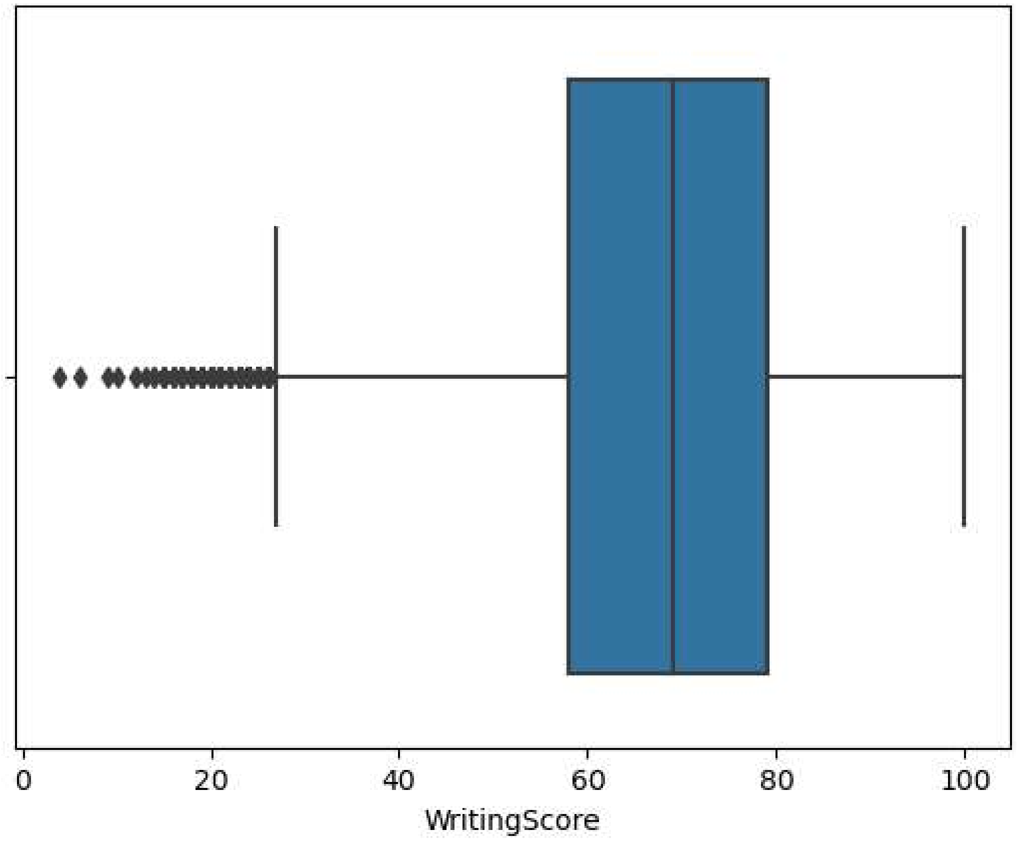
In [55]:

```
sns.boxplot (data = df , x = "ReadingScore")
plt.show()
```



In [56]:

```
sns.boxplot (data = df , x = "WritingScore")
plt.show()
```



In [ ]:

```
WE CAN CINCLUDE THAT
MATHS IS HARDER SUBJECT TO SCORE THAN OTHER SUBJECTS
```

In [59]:

```
print(df["EthnicGroup"].unique())
```

```
[nan 'group C' 'group B' 'group A' 'group D' 'group E']
```

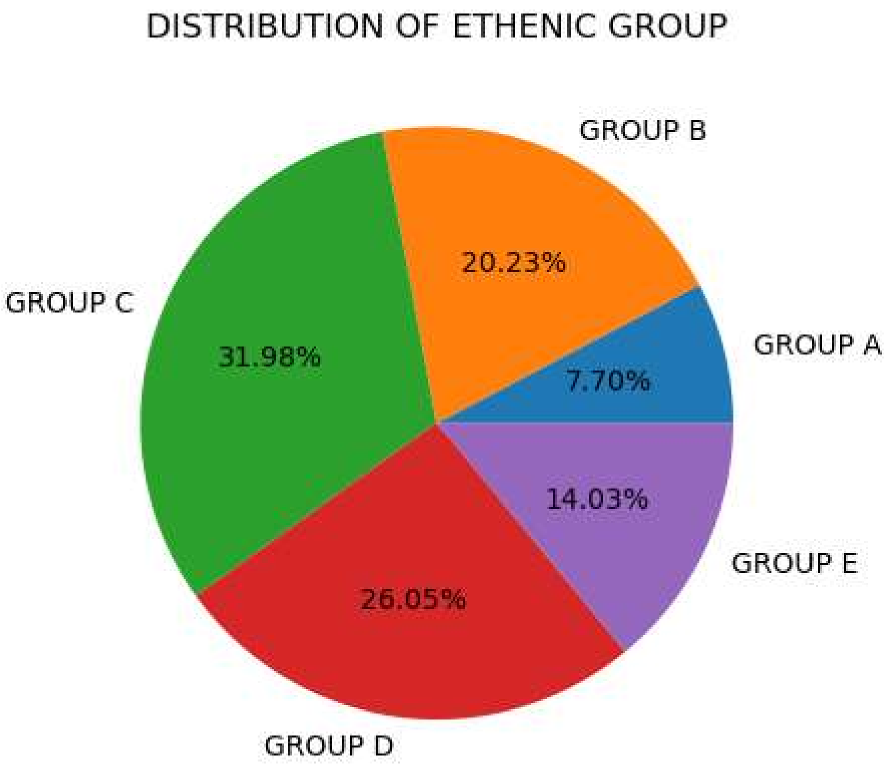
Distribution of ethernic group

In [84]:

```
groupA = df.loc[(df["EthnicGroup"] == "group A")].count()
groupB = df.loc[(df["EthnicGroup"] == "group B")].count()
groupC = df.loc[(df["EthnicGroup"] == "group C")].count()
groupD = df.loc[(df["EthnicGroup"] == "group D")].count()
groupE = df.loc[(df["EthnicGroup"] == "group E")].count()

l= ["GROUP A" , "GROUP B" , "GROUP C" , "GROUP D" , "GROUP E"]
mlist = [groupA["EthnicGroup"], groupB["EthnicGroup"] , groupC["EthnicGrou
print(mlist)
plt.pie(mlist,labels = l, autopct = "%1.2f%%" )
plt.title ("DISTRIBUTION OF ETHENIC GROUP")
plt.show()
```

```
[2219, 5826, 9212, 7503, 4041]
```


DISTRIBUTION OF ETHENIC GROUP
GROUP B
GROUP C
GROUP A
GROUP E
GROUP D