```
#Hasan/lenovo/pc
#Student data

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

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

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

```
df.describe()
```

```
         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
```

```
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
```

```
df.isnull().sum()
```

```
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

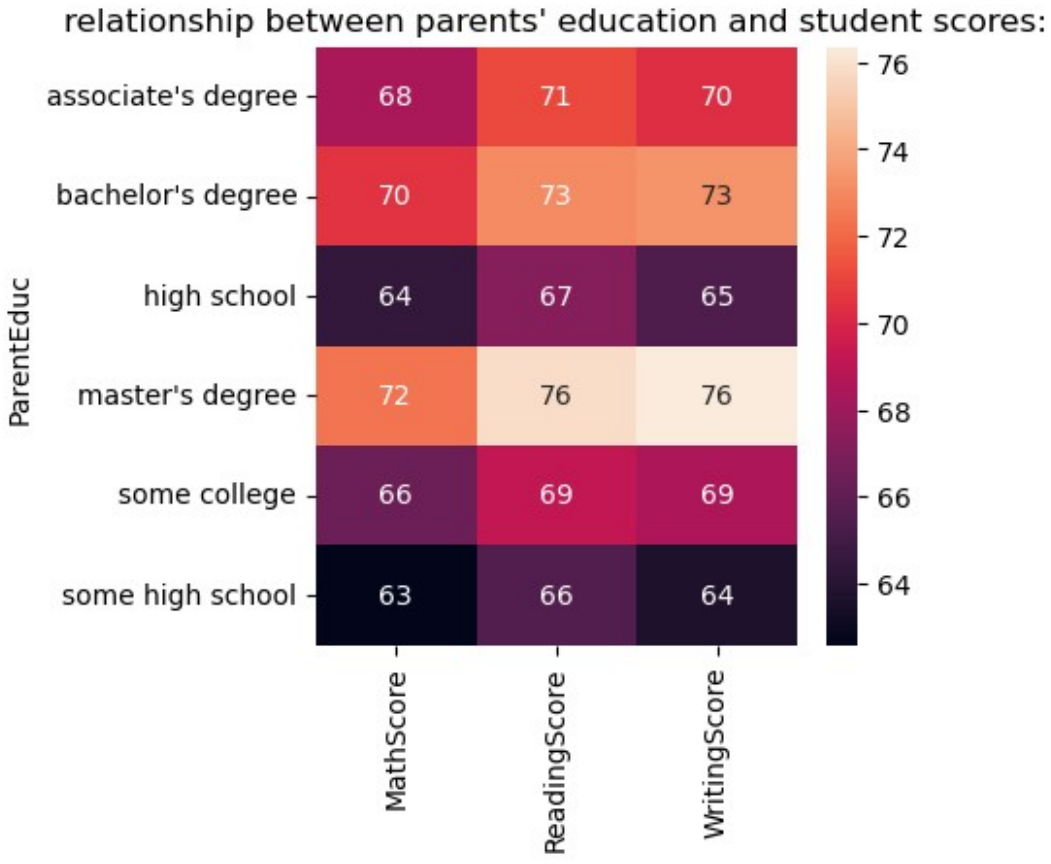
```
plt.figure(figsize=(5,5))
ax = sns.countplot(data=df, x="Gender")
ax.bar_label(ax.containers[0])
plt.title("Gender Distribution")
plt.show()
```


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

```
print(gb)
```

ParentEduc	MathScore	ReadingScore	WritingScore
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

```
plt.figure(figsize=(4,4))
sns.heatmap(gb,annot = True)
plt.title("relationship between parents' education and student
scores:")
plt.show()
```



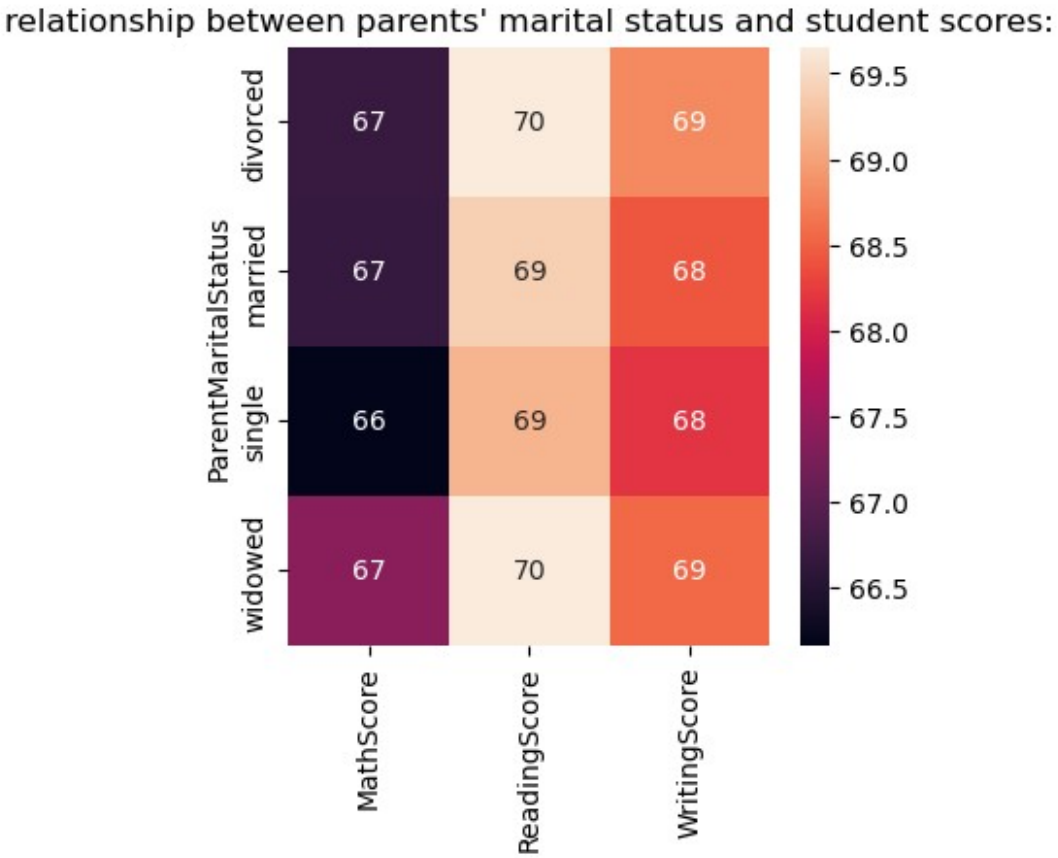
#From the chart, we can conclude that parental qualifications have an impact on student performance.

```
gb1 = df.groupby("ParentMaritalStatus").agg({"MathScore" :
'mean',"ReadingScore" : 'mean',"WritingScore" : 'mean'})
```

```
print(gb1)
```

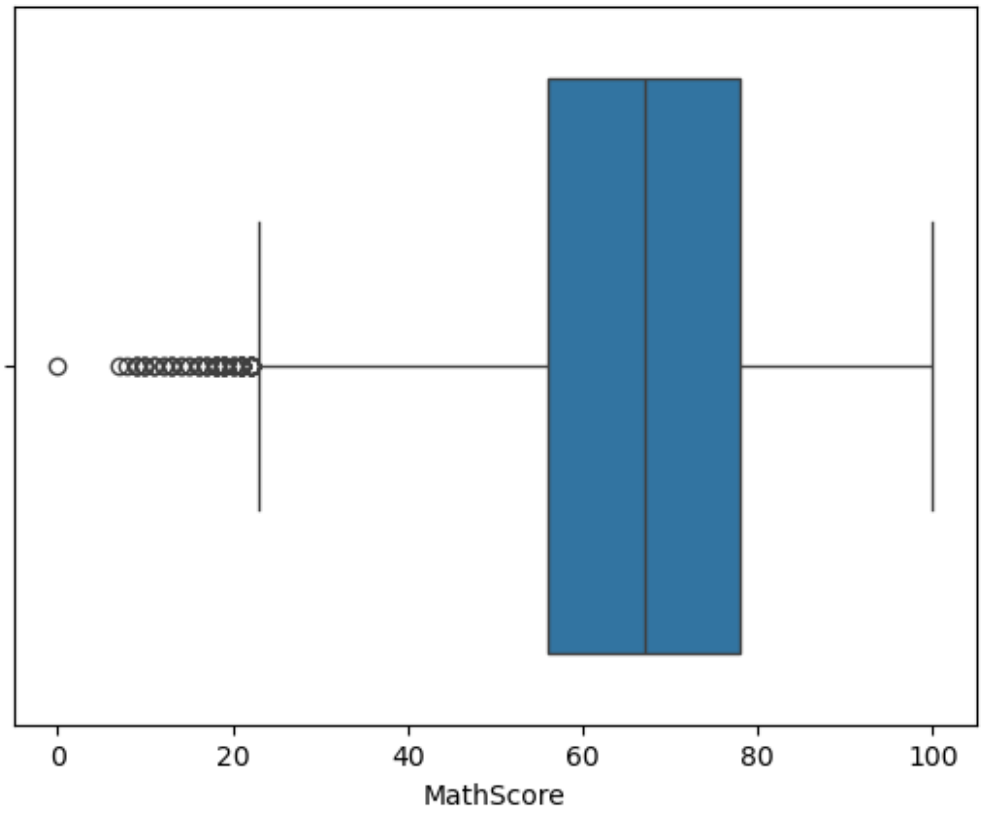
ParentMaritalStatus	MathScore	ReadingScore	WritingScore
divorced	66.691197	69.655011	68.799146
married	66.657326	69.389575	68.420981
single	66.165704	69.157250	68.174440
widowed	67.368866	69.651438	68.563452

```
plt.figure(figsize=(4,4))
sns.heatmap(gb1,annot = True)
plt.title("relationship between parents' marital status and student scores:")
plt.show()
```

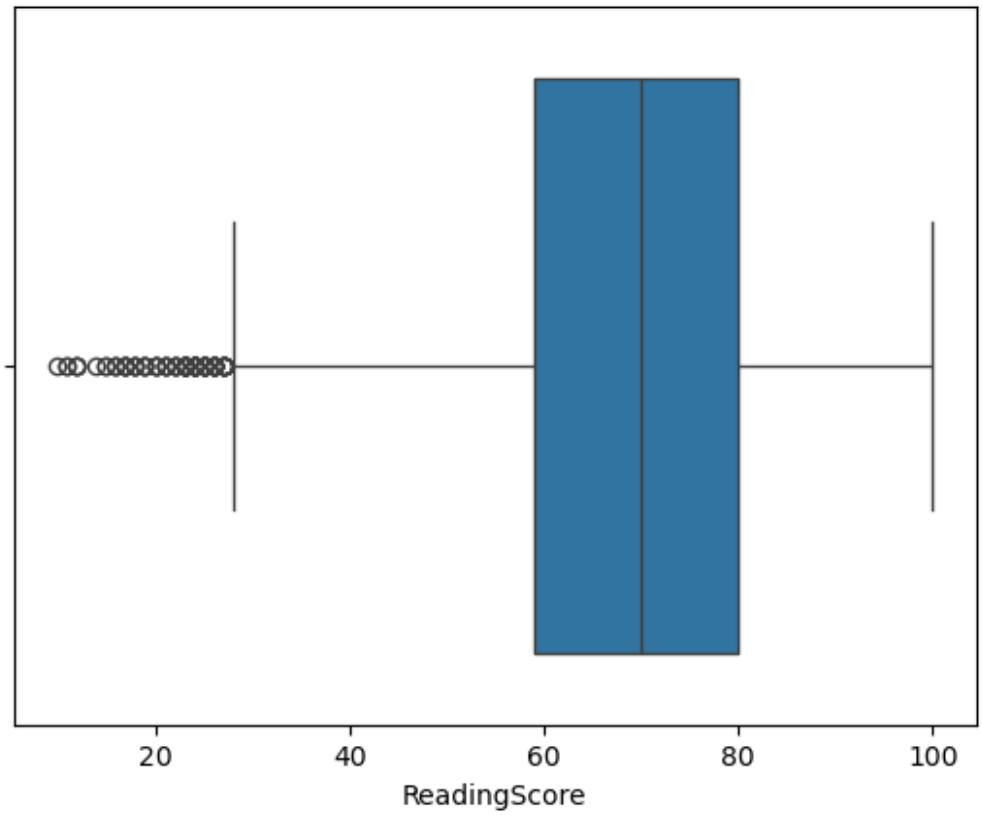


#From the chart above, we can observe that marital status does not have a significant impact on student marks.

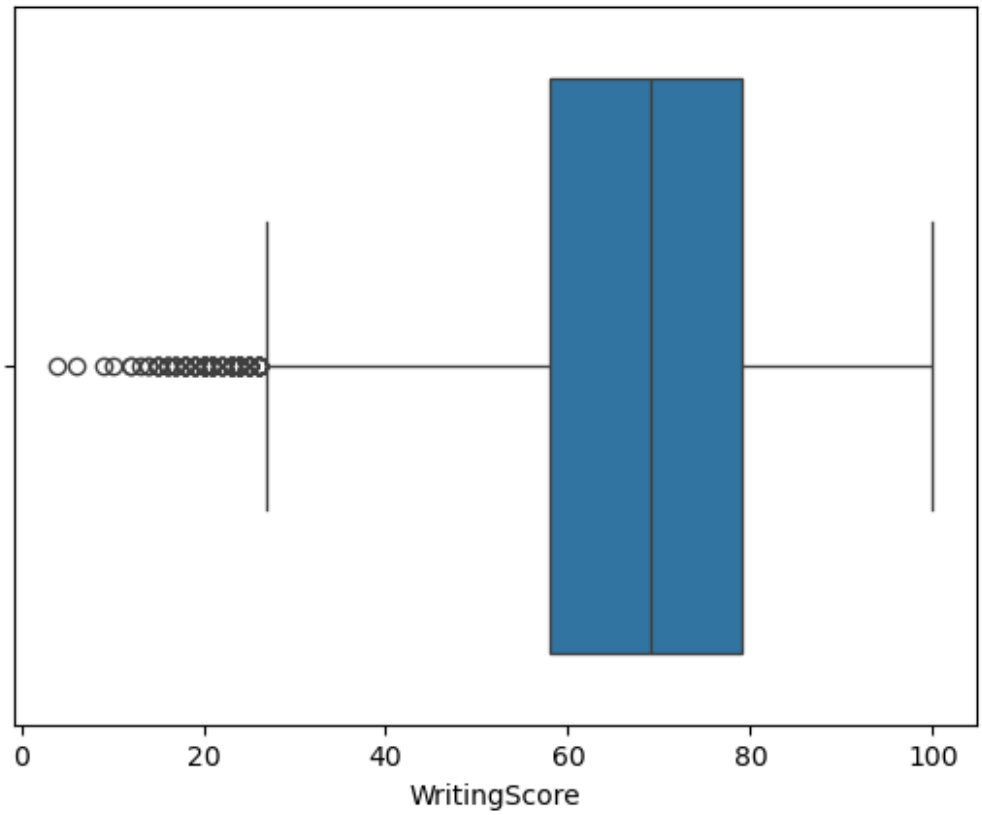
```
sns.boxplot(data = df, x="MathScore")
plt.show()
```




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




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




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

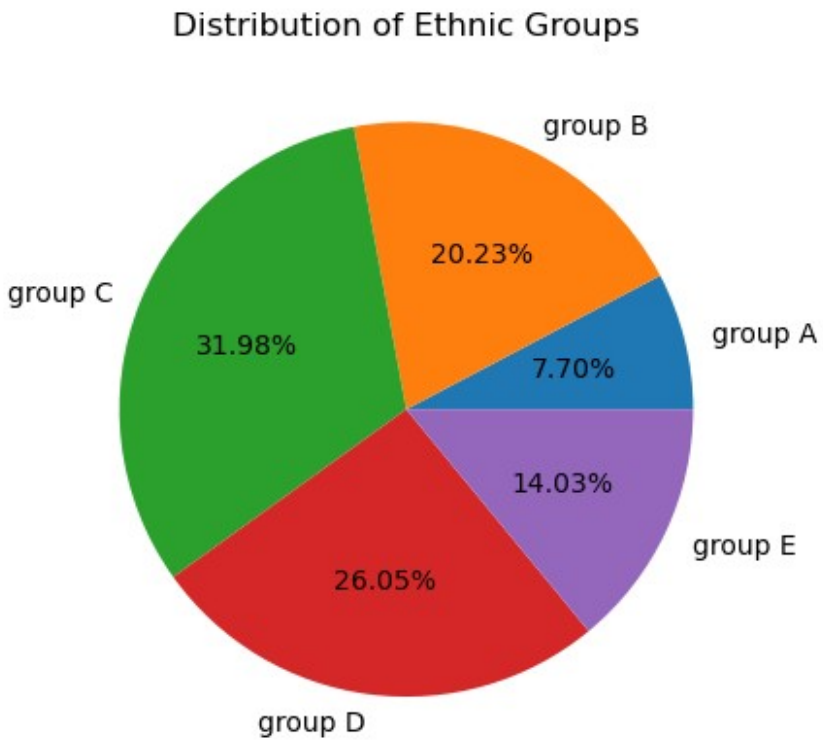
```
[nan 'group C' 'group B' 'group A' 'group D' 'group E']
```

#Distribution of Ethnic Groups

```
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["EthnicGroup"], groupD["EthnicGroup"], groupE["EthnicGroup"]]

plt.pie(mlist, labels=l, autopct="%1.2f%%")
plt.title("Distribution of Ethnic Groups")
plt.show()
```

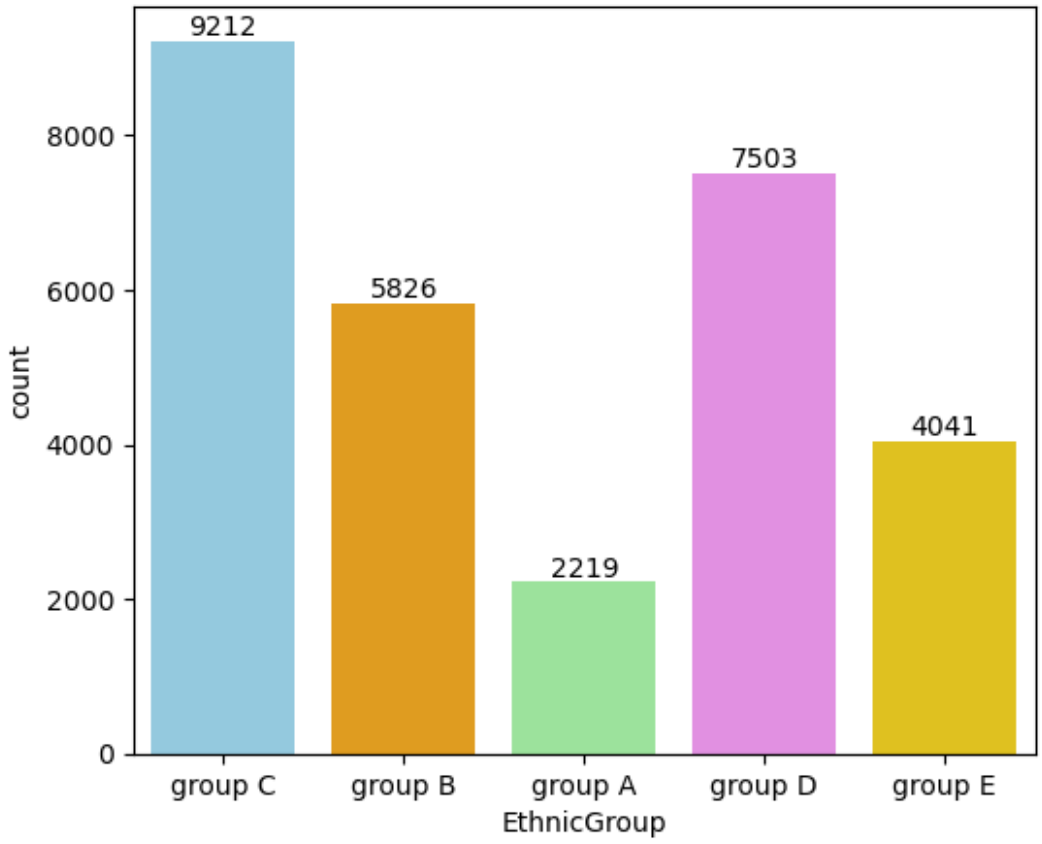



```
plt.figure(figsize=(6, 5))
custom_colors = ["skyblue", "orange", "lightgreen", "violet", "gold"]

ax = sns.countplot(data=df, x='EthnicGroup', hue='EthnicGroup',
palette=custom_colors, legend=False)

for container in ax.containers:
    ax.bar_label(container)

plt.show()
```


9212
7503
5826
4041
2219
8000
6000
4000
2000
0
count
group C
group B
group A
group D
group E
EthnicGroup