In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
```

In [7]:
```
data=pd.read_csv(r"C:\Users\User\Downloads\PROJECTS\newproject\notebook\data\stud.csv
```

In [8]:
```
data.columns
```

Out[8]:
```
Index(['gender', 'race_ethnicity', 'parental_level_of_education', 'lunch',
       'test_preparation_course', 'math_score', 'reading_score',
       'writing_score'],
      dtype='object')
```

In [9]:
```
data.describe()
```

Out[9]:

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

```
75% of the student scored above 75 marks across all the subjects
marks in math score is as low as o marks out of 100 while the others are below 20
majority of the Students marks flactuate from the mean of 65 by 15 marks
```

In [30]:
```
listcol=data.columns
list
```

Out[30]:
```
Index(['gender', 'race_ethnicity', 'parental_level_of_education', 'lunch',
       'test_preparation_course', 'math_score', 'reading_score',
       'writing_score'],
      dtype='object')
```

In [83]:
```
"""
create an instance of an empty dataframe to store each catrgorical dataframe
create a list of datframes to cumulative store all the datframes created in one place
"""
race_df=pd.DataFrame()
for col in range(1):
    race_df[col]=data.race_ethnicity.value_counts()

gender_df=pd.DataFrame()
for col in range(1):
    gender_df[col]=data.gender.value_counts()


parentEdu_df=pd.DataFrame()
for col in range(1):
    parentEdu_df[col]=data.parental_level_of_education.value_counts()

lunch_df=pd.DataFrame()
for col in range(1):
    lunch_df[col]=data.lunch.value_counts()

testPre_df=pd.DataFrame()
for col in range(1):
    testPre_df[col]=data.test_preparation_course.value_counts()

list_of_dataframes=[]
list_of_dataframes.extend([gender_df,parentEdu_df,lunch_df,testPre_df])
list_of_dataframes
```

Out[83]:
```
[          0
 female  518
 male    482,
                       0
 some college        226
 associate's degree  222
 high school         196
 some high school    179
 bachelor's degree   118
 master's degree      59,
                  0
 standard       645
 free/reduced   355,
              0
 none       642
 completed  358]
```

```
the occurence of each feature as as shown above
```

In [87]:
```
data.isna().sum()
```

Out[87]:
```
gender                       0
race_ethnicity               0
parental_level_of_education  0
lunch                        0
test_preparation_course      0
math_score                   0
reading_score                0
writing_score                0
dtype: int64
```

there are no null records in the data

In [94]:
```
data['total_marks']=data['math_score']+data['reading_score']+data['writing_score']
subjects=3
data['average']=data['total_marks']/subjects
data.head(5)
```

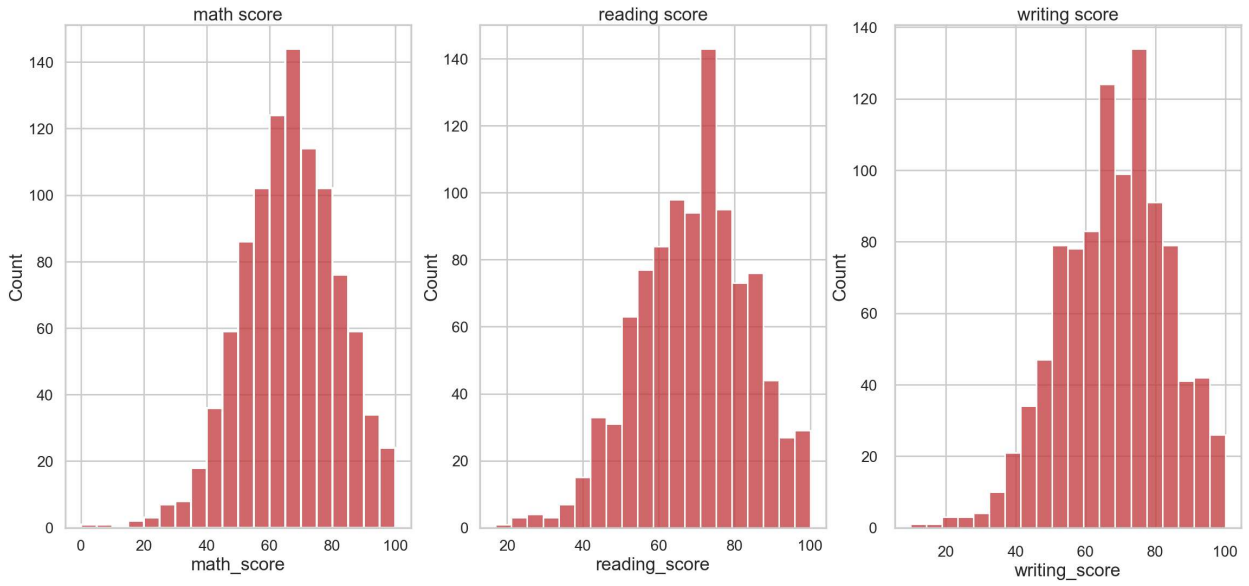
Out[94]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	r
0	female	group B	bachelor's degree	standard	none	72	
1	female	group C	some college	standard	completed	69	
2	female	group B	master's degree	standard	none	90	
3	male	group A	associate's degree	free/reduced	none	47	
4	male	group C	some college	standard	none	76	

VISUALIZATION

In [143]:
```
#distribution of the numeriacal variables
```

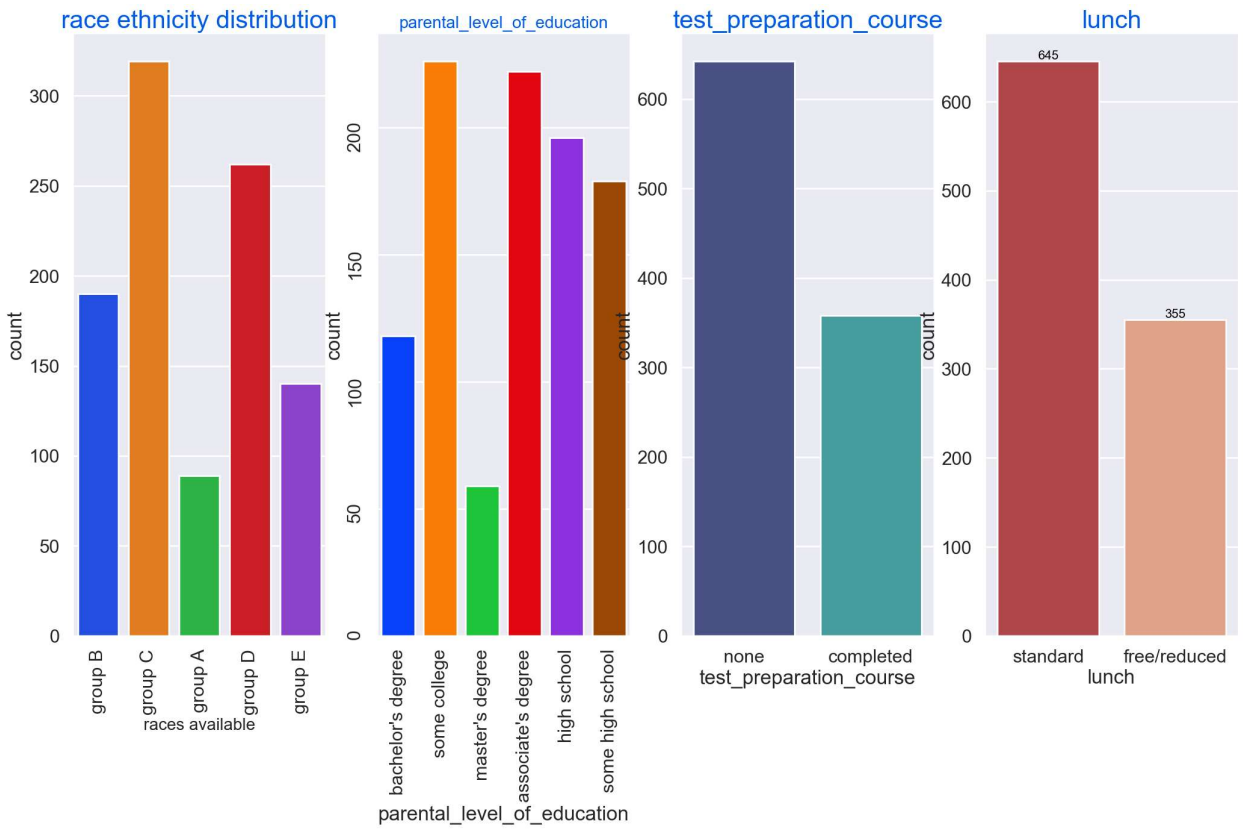
In [313]:
```
fig,ax=plt.subplots(1,3,figsize=(28,12))
sns.set_style('whitegrid')
sns.set_palette('RdBu')

sns.histplot(data=data,x='math_score',ax=ax[0],bins=20,edgecolor='white',linewidth=2.
ax[0].tick_params(axis='x', labelsize=20)
ax[0].tick_params(axis='y', labelsize=20)
ax[0].set_title('math score')

sns.histplot(data=data,x='reading_score',ax=ax[1],bins=20,edgecolor='white')
ax[1].tick_params(axis='x', labelsize=20)
ax[1].tick_params(axis='y', labelsize=20)
ax[1].set_title('reading score')

sns.histplot(data=data,x='writing_score',ax=ax[2],bins=20,edgecolor='white')
ax[2].tick_params(size=20,labelsize=20)
ax[2].set_title('writing score')

plt.show()
```



60 students scoring 50 and below marks across all the subjects

In [182]:
```
#distribution of the categorical variables
```

In [258]:
```
fig,ax=plt.subplots(1,4,figsize=(25,13))
#sns.set_context("poster", rc={"font.size": 20})#set the context(size of the numbers

sns.countplot(data=data,x='race_ethnicity',ax=ax[0],palette='bright')
ax[0].set_title('race ethnicity distribution',color='#005ce6',size=30)
ax[0].set_xlabel('races available',size=20)
ax[0].tick_params(axis='x',rotation=90)

sns.countplot(data=data,x='parental_level_of_education',ax=ax[1],palette='bright',sat
ax[1].set_title('parental_level_of_education',color='#005ce6',size=20)
ax[1].tick_params(rotation=90)


sns.countplot(data=data,x='test_preparation_course',ax=ax[2],palette='mako')
ax[2].set_title('test_preparation_course',color='#005ce6',size=30)



sns.countplot(data=data,x='lunch',ax=ax[3])
ax[3].set_title('lunch',color='#005ce6',size=30)
for container in ax[3].containers:
    ax[3].bar_label(container,color='black',size=15)

plt.show()
```



In [260]:
```
#the correlation between the numerical variables
```

In [261]:
```
corr_matrix=data.corr()
print(corr_matrix)
```

```
              math_score  reading_score  writing_score  total_marks   average
math_score       1.000000       0.817580       0.802642     0.918746  0.918746
reading_score    0.817580       1.000000       0.954598     0.970331  0.970331
writing_score    0.802642       0.954598       1.000000     0.965667  0.965667
total_marks      0.918746       0.970331       0.965667     1.000000  1.000000
average          0.918746       0.970331       0.965667     1.000000  1.000000

C:\Users\User\AppData\Local\Temp\ipykernel_13392\1271945054.py:1: FutureWarning: The
default value of numeric_only in DataFrame.corr is deprecated. In a future version,
it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  corr_matrix=data.corr()
```

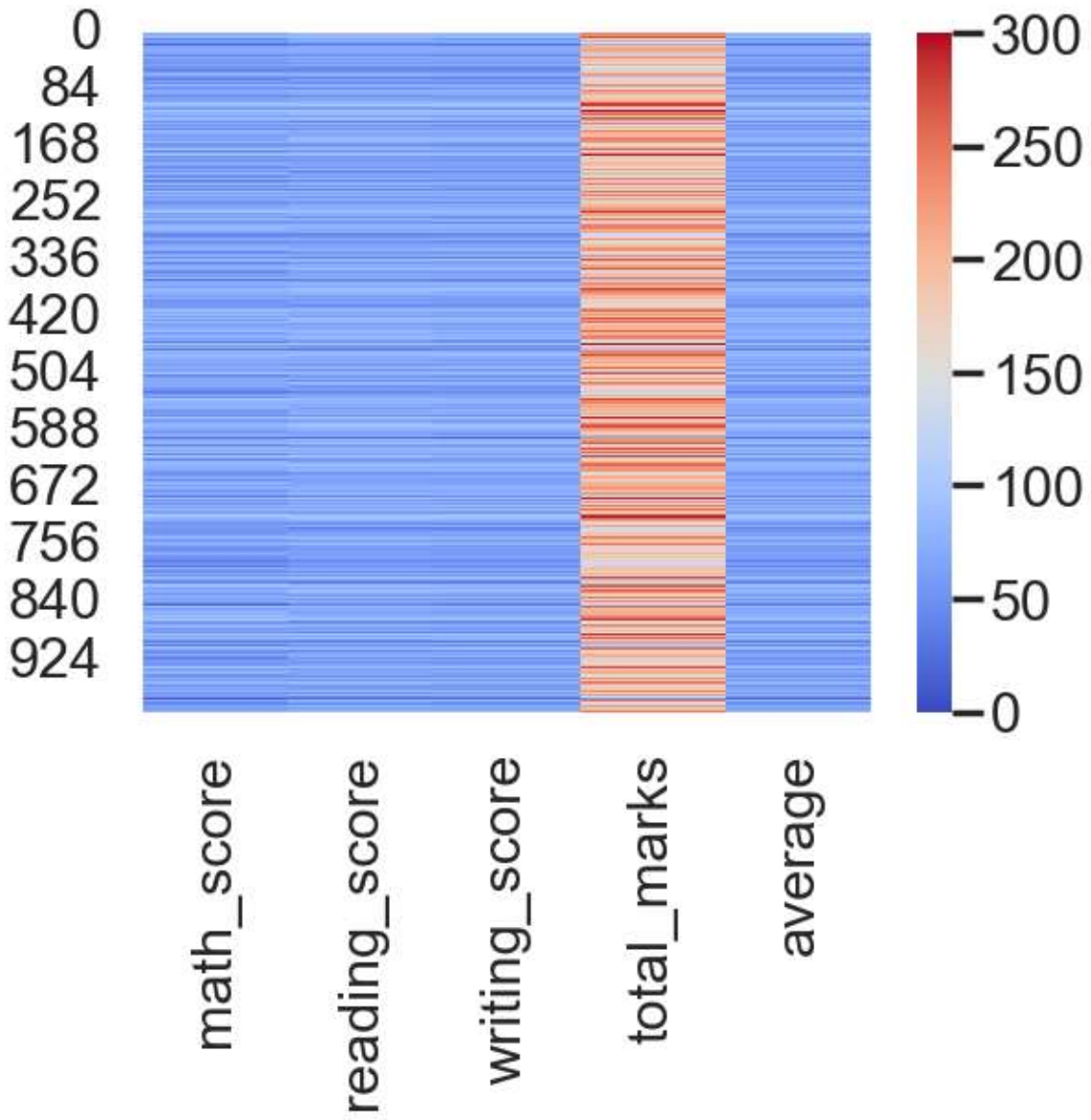
In [ ]:
```
students performing better in maths are likely to perform better in other subjects as
and vice versa.
```

```
or using the numpy correlation coefficient
```

In [269]:
```
numerical_columns=['math_score',  'reading_score',  'writing_score',  'total_marks',
array=data[numerical_variables].to_numpy()
np.corrcoef(array)
```

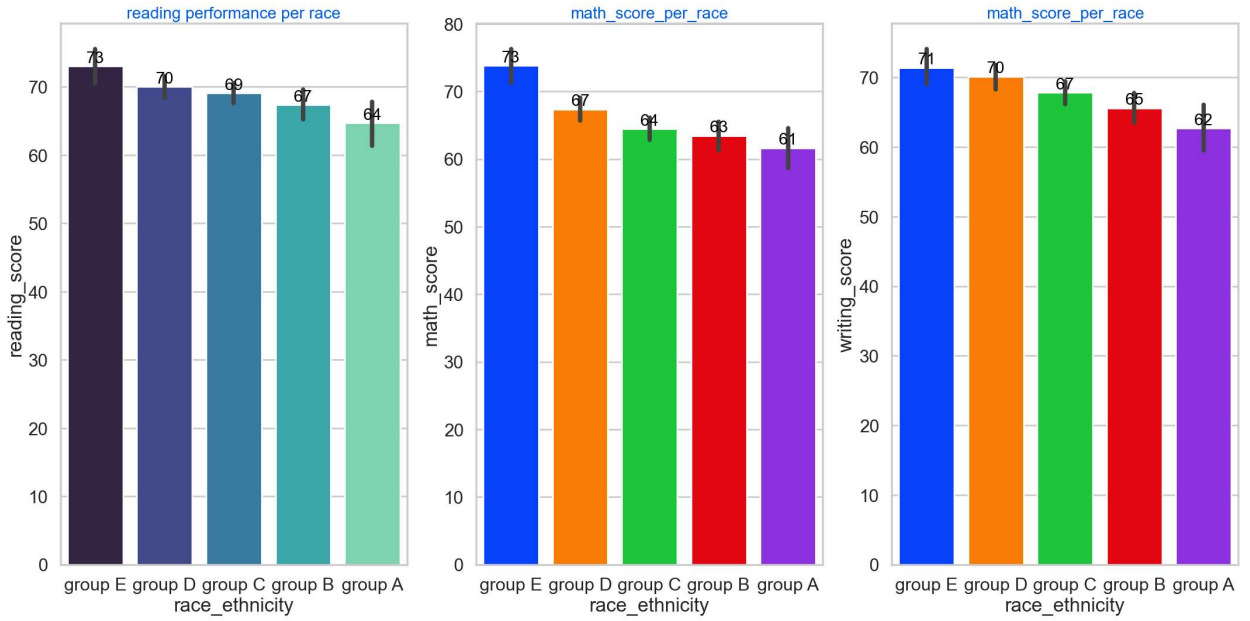
Out[269]:
```
array([[1.        , 0.99437408, 0.99972165, ..., 0.99726918, 0.99851809,
        0.99900914],
       [0.99437408, 1.        , 0.9958872 , ..., 0.99852133, 0.99863128,
        0.99809637],
       [0.99972165, 0.9958872 , 1.        , ..., 0.99867004, 0.9992512 ,
        0.99952105],
       ...,
       [0.99726918, 0.99852133, 0.99867004, ..., 1.        , 0.99950499,
        0.99924436],
       [0.99851809, 0.99863128, 0.9992512 , ..., 0.99950499, 1.        ,
        0.99994706],
       [0.99900914, 0.99809637, 0.99952105, ..., 0.99924436, 0.99994706,
        1.        ]])
```

In [270]:
```
sns.heatmap(data=data[numerical_columns],cmap='coolwarm')
plt.show()
```

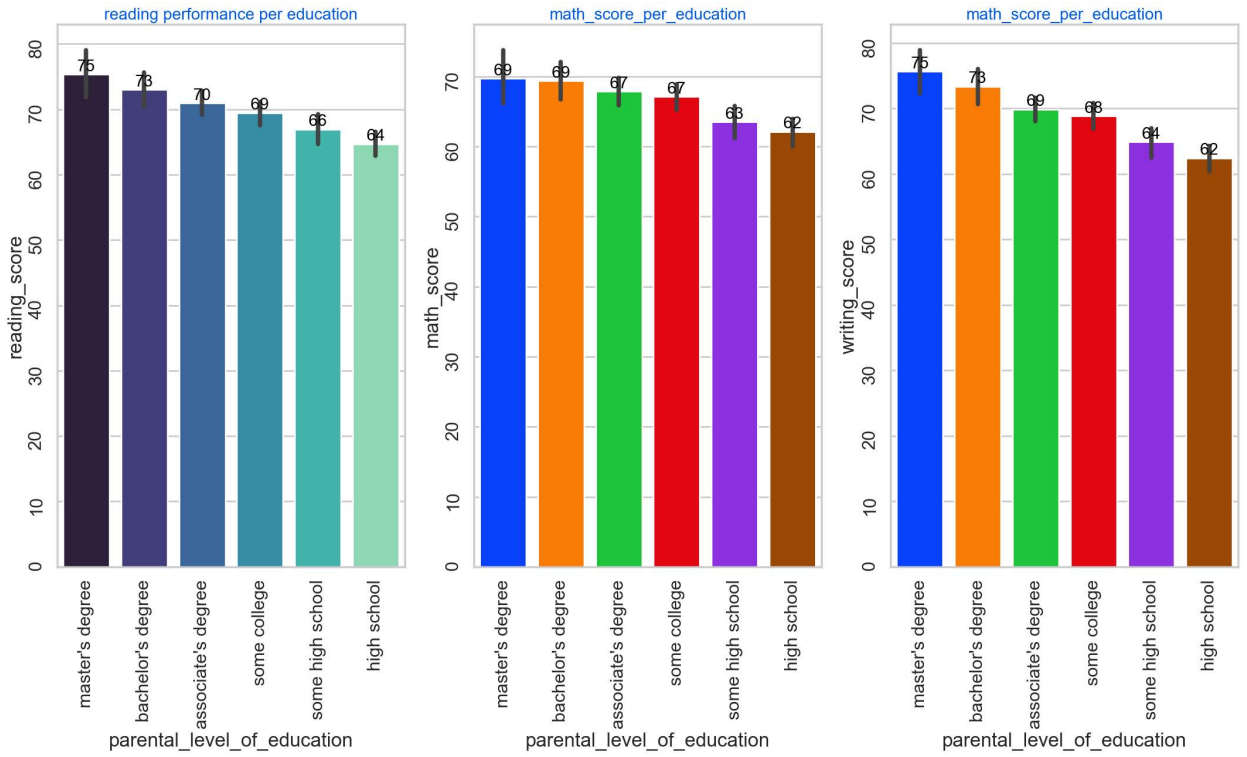


In [271]:
```
#how are the subbjects performed in the different  race groups
```

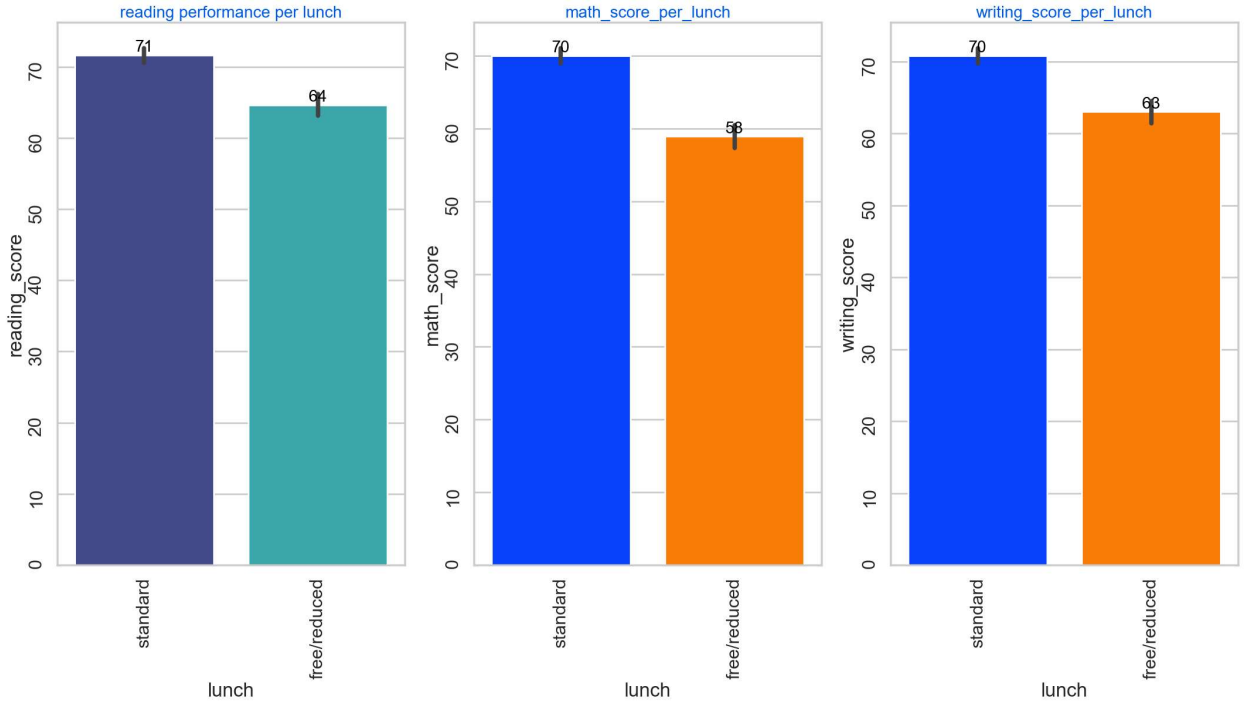
In [298]:
```
fig,ax=plt.subplots(1,3,figsize=(26,12))
order_level=['group E','group D','group C','group B','group A']

sns.barplot(data=data,x='race_ethnicity',y='reading_score',palette='mako',color='whit
ax[0].set_title('reading performance per race',color='#005ce6',size=20)
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='race_ethnicity',y='math_score',palette='bright',color='white
ax[1].set_title('math_score_per_race',color='#005ce6',size=20)
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='race_ethnicity',y='writing_score',palette='bright',color='wh
ax[2].set_title('math_score_per_race',color='#005ce6',size=20)
for container in ax[2].containers:
    ax[2].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')




plt.show()
```



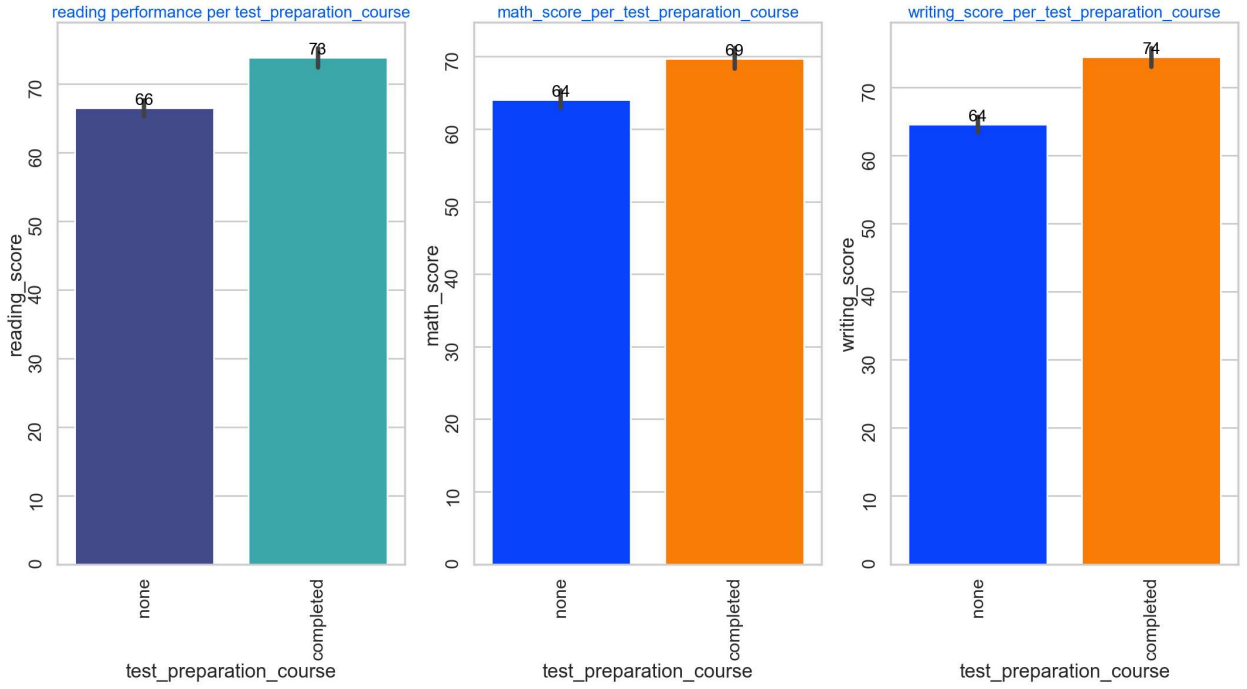
In [307]:
```
fig,ax=plt.subplots(1,3,figsize=(26,12))
order_level=["master's degree","bachelor's degree","associate's degree",'some college

sns.barplot(data=data,x='parental_level_of_education',y='reading_score',palette='mako
ax[0].set_title('reading performance per education',color='#005ce6',size=20)
ax[0].tick_params(rotation=90)
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='parental_level_of_education',y='math_score',palette='bright'
ax[1].set_title('math_score_per_education',color='#005ce6',size=20)
ax[1].tick_params(rotation=90)
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='parental_level_of_education',y='writing_score',palette='brigl
ax[2].set_title('math_score_per_education',color='#005ce6',size=20)
ax[2].tick_params(rotation=90)
for container in ax[2].containers:
    ax[2].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

plt.show()
```



In [310]:
```
fig,ax=plt.subplots(1,3,figsize=(26,12))
order_level=["master's degree","bachelor's degree","associate's degree",'some college


sns.barplot(data=data,x='lunch',y='reading_score',palette='mako',color='white',satura
ax[0].set_title('reading performance per lunch',color='#005ce6',size=20)
ax[0].tick_params(rotation=90)
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='lunch',y='math_score',palette='bright',color='white',saturat
ax[1].set_title('math_score_per_lunch',color='#005ce6',size=20)
ax[1].tick_params(rotation=90)
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='lunch',y='writing_score',palette='bright',color='white',satu
ax[2].set_title('writing_score_per_lunch',color='#005ce6',size=20)
ax[2].tick_params(rotation=90)
for container in ax[2].containers:
    ax[2].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

plt.show()
```



In [311]:
```
fig,ax=plt.subplots(1,3,figsize=(26,12))
order_level=["master's degree","bachelor's degree","associate's degree",'some college


sns.barplot(data=data,x='test_preparation_course',y='reading_score',palette='mako',co
ax[0].set_title('reading performance per test_preparation_course',color='#005ce6',siz
ax[0].tick_params(rotation=90)
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='test_preparation_course',y='math_score',palette='bright',col
ax[1].set_title('math_score_per_test_preparation_course',color='#005ce6',size=20)
ax[1].tick_params(rotation=90)
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

sns.barplot(data=data,x='test_preparation_course',y='writing_score',palette='bright',
ax[2].set_title('writing_score_per_test_preparation_course',color='#005ce6',size=20)
ax[2].tick_params(rotation=90)
for container in ax[2].containers:
    ax[2].bar_label(container,color='black',size=20,fmt='%d', label_type='edge')

plt.show()
```



In [ ]:
```
completing test course results in higher performance in reading than math and writing
coompleting the tests results in better performance than not completing the tests
```