In [1]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,classification_report
```

In [2]:

```
df=pd.read_csv(r'C:\Users\LENOVO\StudentsPerformance.csv')
```

In [3]:

```
df
```

Out[3]:

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| **0** | female | group B | bachelor's degree | standard | none | 72 | 72 | 74 |
| **1** | female | group C | some college | standard | completed | 69 | 90 | 88 |
| **2** | female | group B | master's degree | standard | none | 90 | 95 | 93 |
| **3** | male | group A | associate's degree | free/reduced | none | 47 | 57 | 44 |
| **4** | male | group C | some college | standard | none | 76 | 78 | 75 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **995** | female | group E | master's degree | standard | completed | 88 | 99 | 95 |
| **996** | male | group C | high school | free/reduced | none | 62 | 55 | 55 |
| **997** | female | group C | high school | free/reduced | completed | 59 | 71 | 65 |
| **998** | female | group D | some college | standard | completed | 68 | 78 | 77 |
| **999** | female | group D | some college | free/reduced | none | 77 | 86 | 86 |

1000 rows × 8 columns

In [4]:

```
df.head(5)
```

Out[4]:

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| **0** | female | group B | bachelor's degree | standard | none | 72 | 72 | 74 |
| **1** | female | group C | some college | standard | completed | 69 | 90 | 88 |
| **2** | female | group B | master's degree | standard | none | 90 | 95 | 93 |
| **3** | male | group A | associate's degree | free/reduced | none | 47 | 57 | 44 |
| **4** | male | group C | some college | standard | none | 76 | 78 | 75 |

In [5]:

```
df.isna().sum()
```

Out[5]:

```
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
```

In [6]:

```
df.shape
```

Out[6]:

```
(1000, 8)
```

In [11]:

```
X=df.drop(['race/ethnicity','parental level of education','lunch','test preparation course'
Y=df['gender']
```

In [12]:

```
class Decision():
    def __init__(self,df,X,Y):
        self.df=df
        self.X=X
        self.Y=Y

    def decision(self):
        self.X=X
        self.Y=Y
        X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
        decission=DecisionTreeClassifier()
        decission.fit(X_train,Y_train)
        Y_pred=decission.predict(X_test)
        print(f"decision tree accuracy_score : {accuracy_score(Y_test,Y_pred)*100}%\n")
        print(f"classification_report\n{classification_report(Y_test,Y_pred)}\n")

    def KNN(self):
        X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
        knn=KNeighborsClassifier()
        knn.fit(X_train,Y_train)
        Y_pred=knn.predict(X_test)
        print(f"\nKNN_accuracy_score : {accuracy_score(Y_test,Y_pred)*100}%\n",)
        print(f"\nclassification_report\n{classification_report(Y_test,Y_pred)}")

D_obj=Decision(df,X,Y)
D_obj.decision()
D_obj.KNN()
```

```
decision tree accuracy_score : 80.5%

classification_report
              precision    recall  f1-score   support

      female       0.83      0.78      0.80       103
        male       0.78      0.84      0.81        97

    accuracy                           0.81       200
   macro avg       0.81      0.81      0.80       200
weighted avg       0.81      0.81      0.80       200



KNN_accuracy_score : 84.0%


classification_report
              precision    recall  f1-score   support

      female       0.86      0.82      0.84       102
        male       0.82      0.86      0.84        98

    accuracy                           0.84       200
   macro avg       0.84      0.84      0.84       200
weighted avg       0.84      0.84      0.84       200
```

In [ ]:

```
```