Navttac Mid Exam

Question 1

In [113]:
```
a=5

for i in range(1,a+1):
    print("*" * i)
    if i == a:
        for z in range(a-1,0,-1):
            print("*" * z)
```

```
*
**
***
****
*****
****
***
**
*
```

Question 2

In [81]:
```
import pandas as pd


df = pd.DataFrame({"YearOfBirth":[1930,1940,1950,1965,1973,1982,1987,1992,2010],
                   "LifeExpantancy":[59.7,62.9,70.2,69.2,71.4,74.5,75,75.7,78.7]})
df
```

Out[81]:

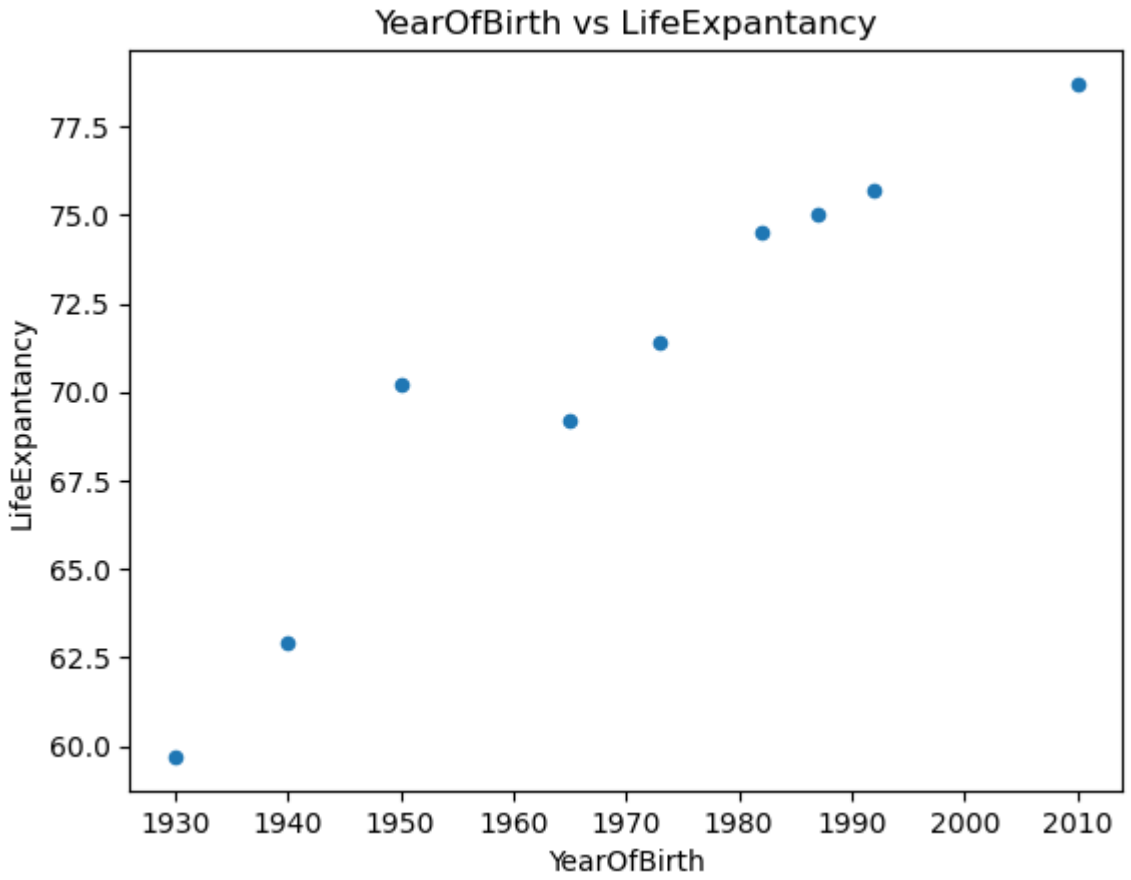
	YearOfBirth	LifeExpantancy
0	1930	59.7
1	1940	62.9
2	1950	70.2
3	1965	69.2
4	1973	71.4
5	1982	74.5
6	1987	75.0
7	1992	75.7
8	2010	78.7

In [82]:
```
# a. Decide dependent and independent

## Ans. YearOfBirth is independent & LifeExpantancy is dependent
```

In [83]:
```
## b. scatter plot
import matplotlib.pyplot as plt

df.plot(kind="scatter",x="YearOfBirth",y="LifeExpantancy")
plt.title("YearOfBirth vs LifeExpantancy")
plt.show()
```



In [84]:
```
## c. find out linear Regression model
import numpy as np
from sklearn.metrics import mean_absolute_error,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

x = np.array(df["YearOfBirth"]).reshape(-1,1)
y = np.array(df["LifeExpantancy"])

x_train ,x_test ,y_train ,y_test = train_test_split(x,y,test_size=0.2 ,random_state=42)
x_train ,x_test ,y_train ,y_test
```

Out[84]:
```
(array([[1982],
        [1930],
        [2010],
        [1950],
        [1973],
        [1965],
        [1987]], dtype=int64),
 array([[1992],
        [1940]], dtype=int64),
 array([74.5, 59.7, 78.7, 70.2, 71.4, 69.2, 75. ]),
 array([75.7, 62.9]))
```

In [85]:
```
model = LinearRegression()
model.fit(x_train , y_train)
```

Out[85]:
```
▾ LinearRegression

LinearRegression()
```

In [86]:
```
y_preds = model.predict(x_test)
y_preds
y_test,y_preds
```

Out[86]:
```
(array([75.7, 62.9]), array([75.86958128, 64.41293103]))
```

In [87]:
```
mean_absolute_error(y_test ,y_preds)
```

Out[87]:
```
0.8412561576354669
```

In [88]:
```
## d.
DateOfBirth = 2002
mylifeExpentancy = model.predict([[DateOfBirth]])
print("My life expantancy is ",mylifeExpentancy)
```

```
My life expantancy is  [78.07278325]
```

In [ ]:
```

```

In [89]:
```
need_tobe_predicted = np.array(range(1930,2011,5)).reshape(-1,1)
need_tobe_predicted
```

Out[89]:
```
array([[1930],
       [1935],
       [1940],
       [1945],
       [1950],
       [1955],
       [1960],
       [1965],
       [1970],
       [1975],
       [1980],
       [1985],
       [1990],
       [1995],
       [2000],
       [2005],
       [2010]])
```

In [90]:
```
y_need_pred = model.predict(need_tobe_predicted)
y_need_pred
```

Out[90]:
```
array([62.20972906, 63.31133005, 64.41293103, 65.51453202, 66.616133  ,
       67.71773399, 68.81933498, 69.92093596, 71.02253695, 72.12413793,
       73.22573892, 74.3273399 , 75.42894089, 76.53054187, 77.63214286,
       78.73374384, 79.83534483])
```

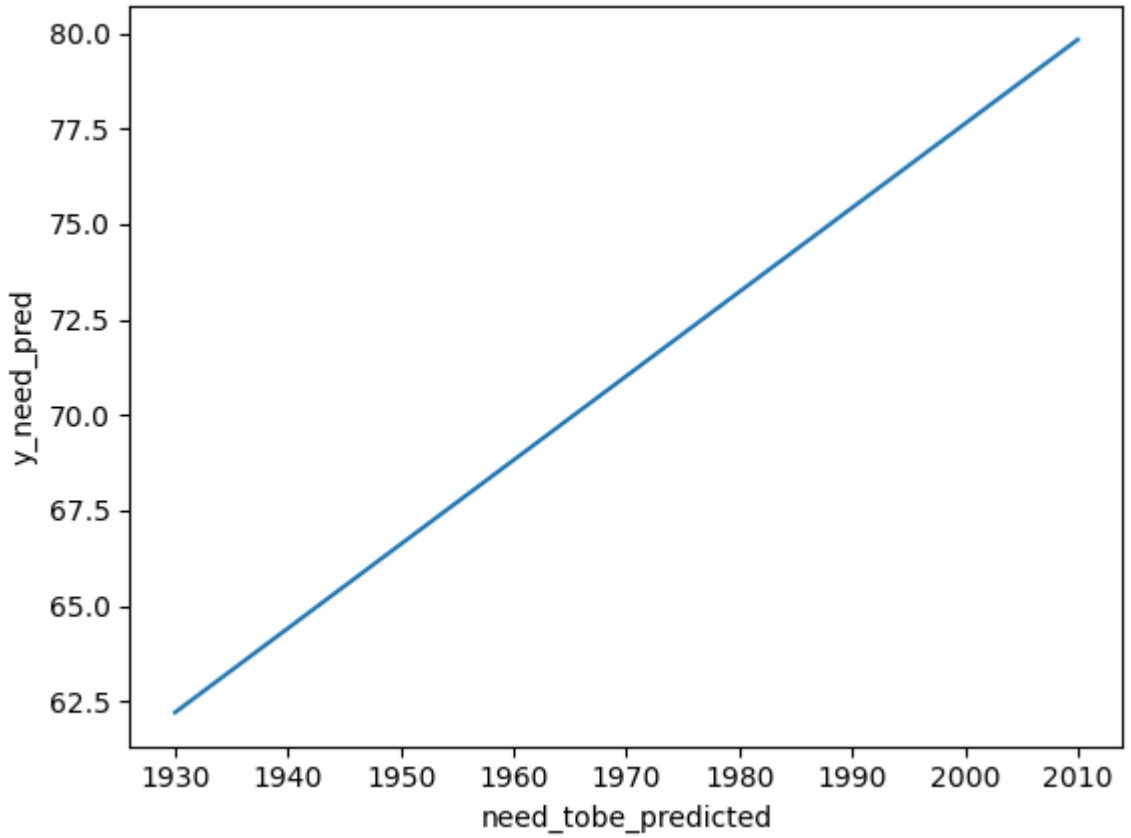
In [91]:
```
dataPred = pd.DataFrame({"need_tobe_predicted":need_tobe_predicted.reshape(1,-1)[0],
                         "y_need_pred":y_need_pred})
dataPred
```

Out[91]:

	need_tobe_predicted	y_need_pred
0	1930	62.209729
1	1935	63.311330
2	1940	64.412931
3	1945	65.514532
4	1950	66.616133
5	1955	67.717734
6	1960	68.819335
7	1965	69.920936
8	1970	71.022537
9	1975	72.124138
10	1980	73.225739
11	1985	74.327340
12	1990	75.428941
13	1995	76.530542
14	2000	77.632143
15	2005	78.733744
16	2010	79.835345

In [92]:
```
## e. plot line plot of predicted values
plt.plot(dataPred["need_tobe_predicted"],dataPred["y_need_pred"]);
plt.xlabel("need_tobe_predicted")
plt.ylabel("y_need_pred")
```

Out[92]:
```
Text(0, 0.5, 'y_need_pred')
```



Question 3

In [121]:
```
names= input("name of Student : ")
rollNo=input("Roll N7 :")

dict = {f"{rollNo}":names}
```

```
name of Student : Ghufran
Roll N7 :key
```

In [122]:
```
#a .dict
dict
```

Out[122]:
```
{'key': 'Ghufran'}
```

In [ ]:
```

```

Question 4

In [103]:
```
exam_data = {"name":["Anastasia","Dima","Kathrine","James","Emily","Micheal","Matthews","Laura","Kelvin","jon
             "score":[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
             "attempts":[1,3,2,3,2,3,1,1,2,1],
             "qualify":["yes","no","yes","no","no","yes","yes","no","no","yes"]}

labels=["a","b","c","d","e","f","g","h","i","j"]

data = pd.DataFrame(exam_data,index=labels)
data
```

Out[103]:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Kathrine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Micheal	20.0	3	yes
g	Matthews	14.5	1	yes
h	Laura	NaN	1	no
i	Kelvin	8.0	2	no
j	jonas	19.0	1	yes

In [118]:
```
data.iloc[0:3]
```

Out[118]:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Kathrine	16.5	2	yes

In [ ]: