Import necessary modules

In [64]:

```
import pandas as pd
import numpy as np
import warnings
```

In [67]:

```
warnings.filterwarnings("ignore")
```

Task1: To generate student IDs for 40 students

In [6]:

```
student_id=pd.Series(index=np.arange(1001,1041),dtype=int)
```

Task2: To generate mathematics marks for 40 students

In [10]:

```
maths_score=pd.Series(np.random.randint(50,95,40),index=np.arange(1001,1041))
```

Task3: To generate physics marks for 40 students

In [11]:

```
physics_score=pd.Series(np.random.randint(50,95,40),index=np.arange(1001,1041))
```

Task4: To find the students who are eligible for level 2 with total marks greater than cut off of 150

In [74]:

```
student_data=pd.DataFrame(maths_score,columns=["Maths1"])
student_data.rename_axis("Student_id")
student_data["Physics1"]=physics_score.values
student_data["Total_score1"]=student_data.Maths1+student_data.Physics1
student_data1=student_data[student_data.Maths1+student_data.Physics1>150]
ran=len(student_data1)
student_data1
```

Out[74]:

	Maths1	Physics1	Total_score1
1002	90	77	167
1005	90	68	158
1010	87	86	173
1011	92	84	176
1015	78	85	163
1022	91	84	175
1028	83	83	166
1029	84	77	161
1031	84	84	168
1033	81	83	164
1035	78	86	164
1037	92	76	168
1038	82	89	171
1039	91	79	170

Task5: To generate maths and physics marks for 40 students who cleared level 1

In [76]:

```
student_data1["Maths2"]=pd.Series(np.random.randint(50,100,ran)).values
student_data1["Physics2"]=pd.Series(np.random.randint(30,100,ran)).values
student_data1
```

Out[76]:

	Maths1	Physics1	Total_score1	Maths2	Physics2
1002	90	77	167	79	82
1005	90	68	158	86	83
1010	87	86	173	66	45
1011	92	84	176	72	86
1015	78	85	163	95	46
1022	91	84	175	66	31
1028	83	83	166	64	52
1029	84	77	161	81	32
1031	84	84	168	85	78
1033	81	83	164	50	79

Task6: To find students who are eligible to move to screening with 300 and more cut off

In [113]:

```
student_level=student_data1[student_data1.Maths1+student_data1.Physics1+student_data1.Maths2+student_data1.Physics2>=300]
student_level["Total_score2"]=student_data1.Maths1+student_data1.Physics1+student_data1.Maths2+student_data1.Physics2
ran=len(student_level)
student_level
```

Out[113]:

	Maths1	Physics1	Total_score1	Maths2	Physics2	Total_score2
1002	90	77	167	79	82	328
1005	90	68	158	86	83	327
1011	92	84	176	72	86	334
1015	78	85	163	95	46	304
1031	84	84	168	85	78	331
1038	82	89	171	88	90	349
1039	91	79	170	88	97	355

Task7: To generate random numbers for maths and physics in screening and take 40% from level2 and add to 60% of screening as total

In [122]:

```
student_level["Maths_sc"]=pd.Series(np.random.randint(50,100,ran)).values
student_level["Physics_sc"]=pd.Series(np.random.randint(30,100,ran)).values
student_level["Final_score"]=(student_level.Maths_sc+student_level.Physics_sc)*0.6+student_level.Total_score2*0.4
student_level
```

Out[122]:

	Maths1	Physics1	Total_score1	Maths2	Physics2	Total_score2	Maths_sc	Physics_sc	Final_score
1002	90	77	167	79	82	328	63	35	190.0
1005	90	68	158	86	83	327	51	85	212.4
1011	92	84	176	72	86	334	89	34	207.4
1015	78	85	163	95	46	304	56	80	203.2
1031	84	84	168	85	78	331	78	75	224.2
1038	82	89	171	88	90	349	60	50	205.6
1039	91	79	170	88	97	355	82	42	216.4

Task8: To find the student who have score more than 75% since its the cutoff in final score

In [130]:

```
eligible_students=student_level[student_level.Final_score>280*0.75]
eligible_students
```

Out[130]:

	Maths1	Physics1	Total_score1	Maths2	Physics2	Total_score2	Maths_sc	Physics_sc	Final_score
1005	90	68	158	86	83	327	51	85	212.4
1031	84	84	168	85	78	331	78	75	224.2
1039	91	79	170	88	97	355	82	42	216.4

Task9: To display in dataframe object and also find highest and lowest scorers amongst eligible students

In [131]:

```
highest=max(eligible_students.Final_score)
lowest=min(eligible_students.Final_score)

print("The highest marks among the eligible students is ->",max(eligible_students.Final_score))
print("The lowest marks among the eligible students is ->",min(eligible_students.Final_score))
```

```
The highest marks among the eligible students is -> 224.2
The lowest marks among the eligible students is -> 212.4
```

The END