INDEX

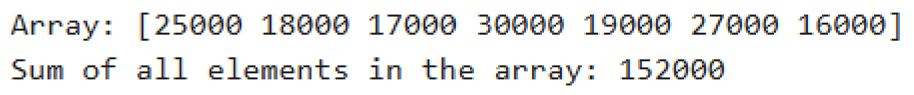
Practical No:01

Aim: Calculate the sum of all elements in the array.

Code:

```
import numpy as np

# Step 1: Create a numeric array (example salaries of employees)

salaries = np.array([25000, 18000, 17000, 30000, 19000, 27000, 16000])


# Step 2: Calculate sum of all elements

total_sum = np.sum(salaries)


# Step 3: Display the result

print("Array:", salaries)

print("Sum of all elements in the array:", total_sum)
```

Output:



```
Array: [25000 18000 17000 30000 19000 27000 16000]
Sum of all elements in the array: 152000
```

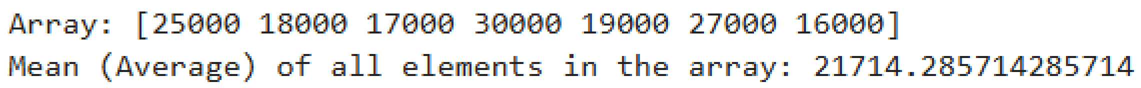
Practical No:02

Aim: Calculate the mean (average) of all elements in the array.

Code:

```
import numpy as np
# Step 1: Create a numeric array
salaries = np.array([25000, 18000, 17000, 30000, 19000, 27000, 16000])
# Step 2: Calculate the mean (average) of all elements
average_salary = np.mean(salaries)
# Step 3: Display the result
print("Array:", salaries)
print("Mean (Average) of all elements in the array:", average_salary)
```

Output:



```
Array: [25000 18000 17000 30000 19000 27000 16000]
Mean (Average) of all elements in the array: 21714.285714285714
```

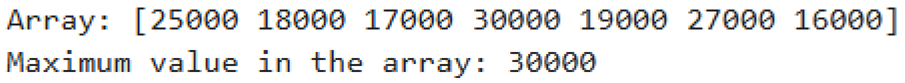
Practical No:03

Aim: Find the maximum value among all elements in the array.

Code:

```
import numpy as np
# Step 1: Create a numeric array
salaries = np.array([25000, 18000, 17000, 30000, 19000, 27000, 16000])
# Step 2: Find the maximum value
max_salary = np.max(salaries)
# Step 3: Display the result
print("Array:", salaries)
print("Maximum value in the array:", max_salary)
```

Output:



```
Array: [25000 18000 17000 30000 19000 27000 16000]
Maximum value in the array: 30000
```

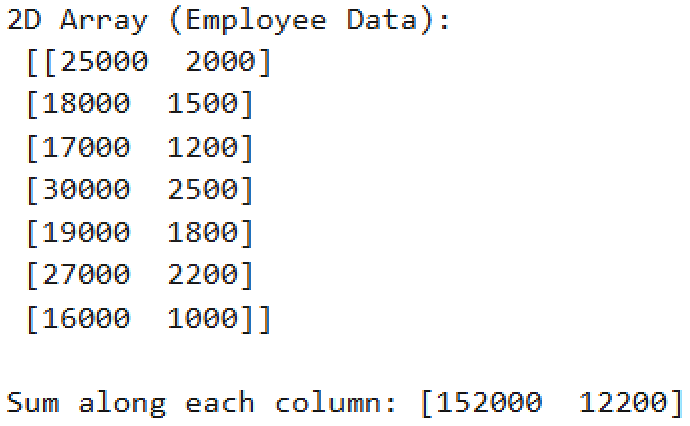
Practical No:04

Aim: Compute the sum of elements along each column (i.e., axis = 0).

Code:

```
import numpy as np
# Step 1: Create a 2D numeric array
# Example: Each row = employee, columns = [Salary, Bonus]
employee_data = np.array([
    [25000, 2000],    [18000, 1500],    [17000, 1200],    [30000, 2500],
    [19000, 1800],    [27000, 2200],    [16000, 1000]
])
# Step 2: Compute sum along each column (axis=0)
column_sum = np.sum(employee_data, axis=0)

# Step 3: Display the result
print("2D Array (Employee Data):\n", employee_data)
print("\nSum along each column:", column_sum)
```

Output:



```
2D Array (Employee Data):
 [[25000  2000]
 [18000  1500]
 [17000  1200]
 [30000  2500]
 [19000  1800]
 [27000  2200]
 [16000  1000]]

Sum along each column: [152000  12200]
```

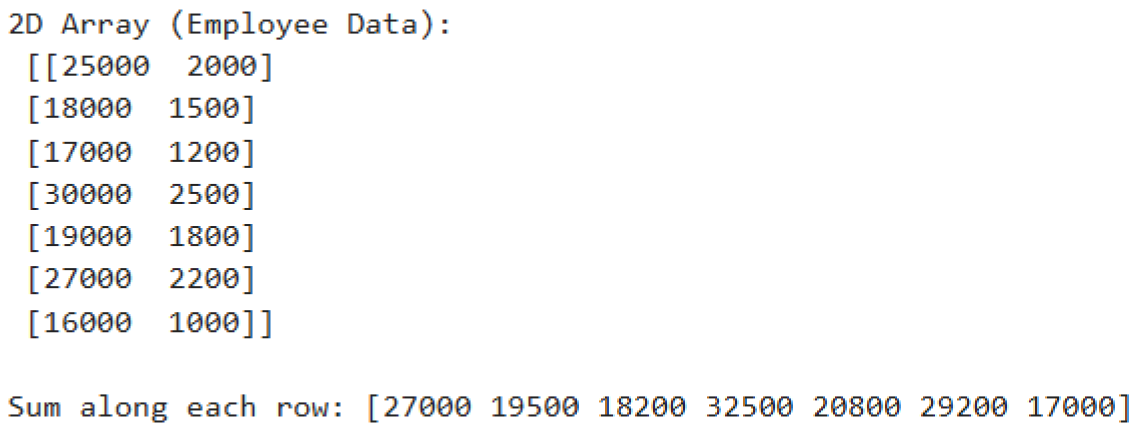
Practical No:05

Aim: Compute the sum of elements along each row (i.e., axis = 1).

Code:

```
import numpy as np
# Step 1: Create a 2D numeric array
# Each row = employee, columns = [Salary, Bonus]
employee_data = np.array([
    [25000, 2000],    [18000, 1500],    [17000, 1200],    [30000, 2500],
    [19000, 1800],    [27000, 2200],    [16000, 1000]
])

# Step 2: Compute sum along each row (axis=1)
row_sum = np.sum(employee_data, axis=1)

# Step 3: Display the result
print("2D Array (Employee Data):\n", employee_data)
print("\nSum along each row:", row_sum)
```

Output:



```
2D Array (Employee Data):
 [[25000  2000]
 [18000  1500]
 [17000  1200]
 [30000  2500]
 [19000  1800]
 [27000  2200]
 [16000  1000]]

Sum along each row: [27000 19500 18200 32500 20800 29200 17000]
```

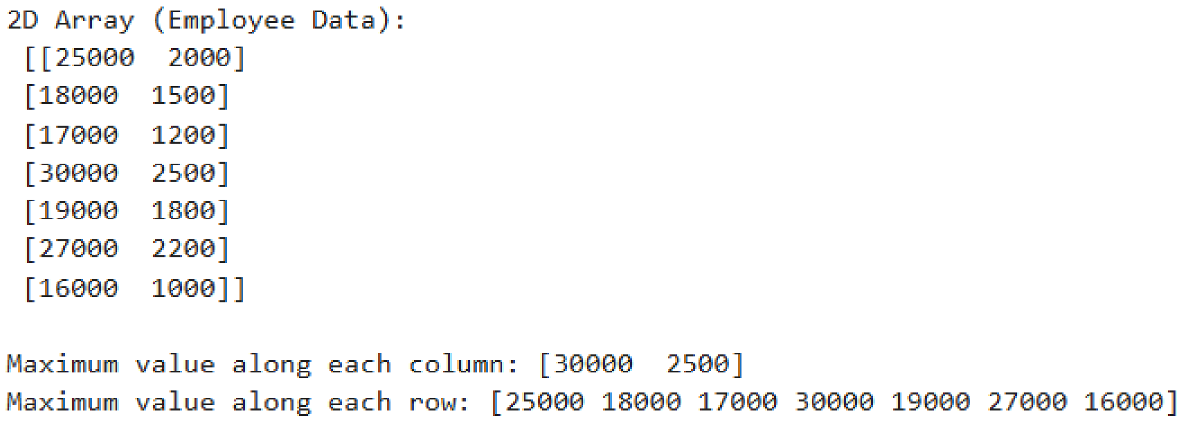
Practical No:06

Aim: Find the maximum value along each column and each row (axis = 0 and axis = 1).

Code:

```
import numpy as np
# Step 1: Create a 2D numeric array
# Each row = employee, columns = [Salary, Bonus]
employee_data = np.array([
    [25000, 2000],    [18000, 1500],    [17000, 1200],    [30000, 2500],
    [19000, 1800],    [27000, 2200],    [16000, 1000]
])
# Step 2: Maximum along each column (axis=0)
max_col = np.max(employee_data, axis=0)
# Step 3: Maximum along each row (axis=1)
max_row = np.max(employee_data, axis=1)
# Step 4: Display results
print("2D Array (Employee Data):\n", employee_data)
print("\nMaximum value along each column:", max_col)
print("Maximum value along each row:", max_row)
```

Output:



```
2D Array (Employee Data):
 [[25000  2000]
 [18000  1500]
 [17000  1200]
 [30000  2500]
 [19000  1800]
 [27000  2200]
 [16000  1000]]

Maximum value along each column: [30000  2500]
Maximum value along each row: [25000 18000 17000 30000 19000 27000 16000]
```

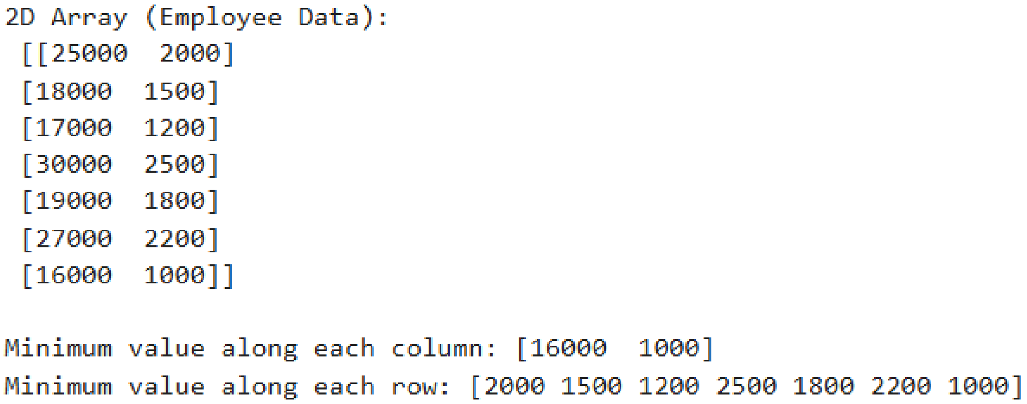
Practical No:07

Aim: Find the minimum value along each column and each row (axis = 0 and axis =1).

Code:

```
import numpy as np
# Step 1: Create a 2D numeric array
# Each row = employee, columns = [Salary, Bonus]
employee_data = np.array([
    [25000, 2000],    [18000, 1500],    [17000, 1200],    [30000, 2500],
    [19000, 1800],    [27000, 2200],    [16000, 1000]
])
# Step 2: Minimum along each column (axis=0)
min_col = np.min(employee_data, axis=0)
# Step 3: Minimum along each row (axis=1)
min_row = np.min(employee_data, axis=1)
# Step 4: Display results
print("2D Array (Employee Data):\n", employee_data)
print("\nMinimum value along each column:", min_col)
print("Minimum value along each row:", min_row)
```

Output:



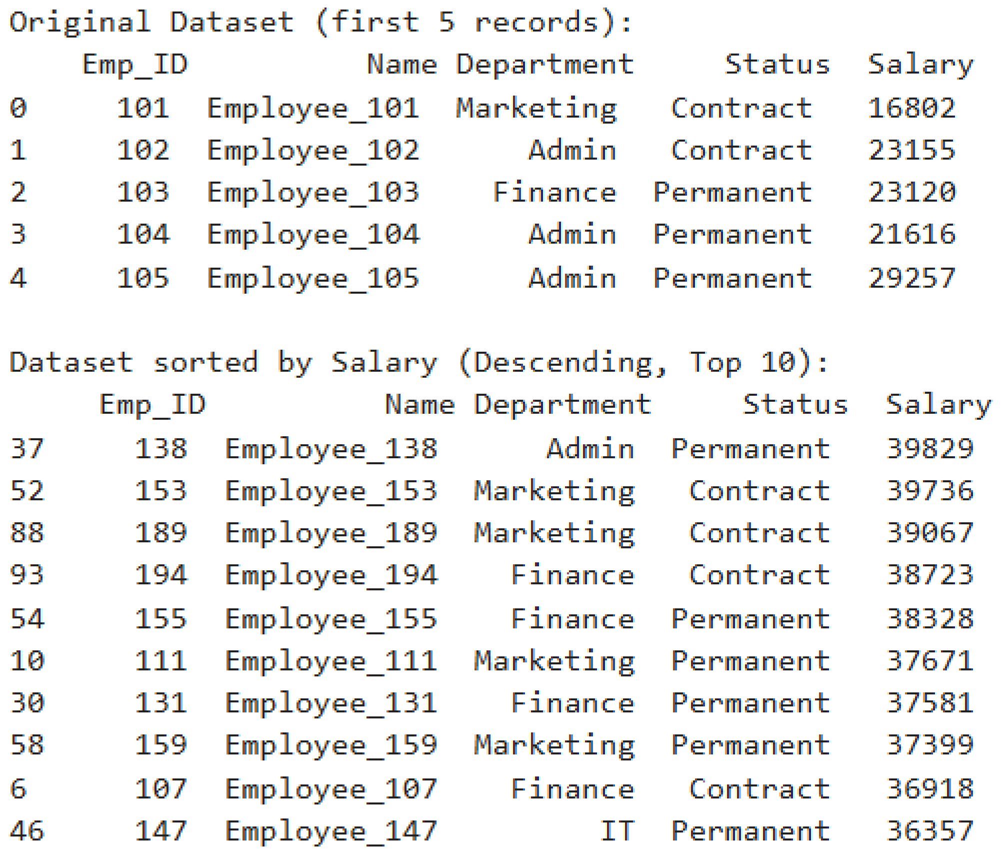
```
2D Array (Employee Data):
 [[25000  2000]
 [18000  1500]
 [17000  1200]
 [30000  2500]
 [19000  1800]
 [27000  2200]
 [16000  1000]]

Minimum value along each column: [16000  1000]
Minimum value along each row: [2000 1500 1200 2500 1800 2200 1000]
```

Practical No:08
Aim: Display the DataFrame sorted in descending order of Salary.

Code:

```
import pandas as pd
# Step 1: Load the dataset
df = pd.read_csv("employee_data.csv")
# Step 2: Display first few records before sorting
print("Original Dataset (first 5 records):\n", df.head())
# Step 3: Sort the DataFrame by Salary in descending order
sorted_df = df.sort_values(by='Salary', ascending=False)
# Step 4: Display the sorted DataFrame
print("\nDataset sorted by Salary (Descending, Top 10):\n", sorted_df.head(10))
```

Output:



```
Original Dataset (first 5 records):
    Emp_ID          Name Department     Status  Salary
0     101  Employee_101  Marketing   Contract   16802
1     102  Employee_102      Admin   Contract   23155
2     103  Employee_103    Finance  Permanent   23120
3     104  Employee_104      Admin  Permanent   21616
4     105  Employee_105      Admin  Permanent   29257

Dataset sorted by Salary (Descending, Top 10):
     Emp_ID          Name Department     Status  Salary
37     138  Employee_138      Admin  Permanent   39829
52     153  Employee_153  Marketing   Contract   39736
88     189  Employee_189  Marketing   Contract   39067
93     194  Employee_194    Finance   Contract   38723
54     155  Employee_155    Finance  Permanent   38328
10     111  Employee_111  Marketing  Permanent   37671
30     131  Employee_131    Finance  Permanent   37581
58     159  Employee_159  Marketing  Permanent   37399
6      107  Employee_107    Finance   Contract   36918
46     147  Employee_147         IT  Permanent   36357
```

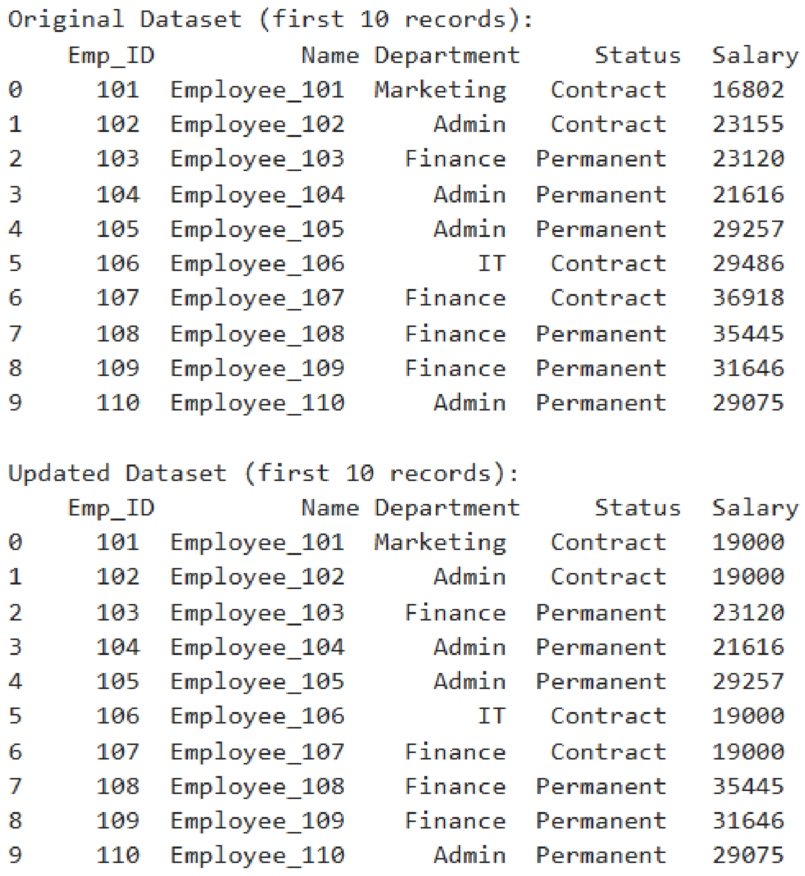
Practical No:09

Aim: Update the Salary of all employees with ‘Contract’ status to ₹19,000.

Code:

```
import pandas as pd
# Step 1: Load the dataset
df = pd.read_csv("employee_data.csv")
# Step 2: Display first few records before update
print("Original Dataset (first 10 records):\n", df.head(10))
# Step 3: Update Salary for Contract employees
df.loc[df['Status'] == 'Contract', 'Salary'] = 19000
# Step 4: Display the updated DataFrame
print("\nUpdated Dataset (first 10 records):\n", df.head(10))
```

Output:



```
Original Dataset (first 10 records):
    Emp_ID          Name Department     Status  Salary
0      101  Employee_101  Marketing   Contract   16802
1      102  Employee_102      Admin   Contract   23155
2      103  Employee_103    Finance  Permanent   23120
3      104  Employee_104      Admin  Permanent   21616
4      105  Employee_105      Admin  Permanent   29257
5      106  Employee_106         IT   Contract   29486
6      107  Employee_107    Finance   Contract   36918
7      108  Employee_108    Finance  Permanent   35445
8      109  Employee_109    Finance  Permanent   31646
9      110  Employee_110      Admin  Permanent   29075

Updated Dataset (first 10 records):
    Emp_ID          Name Department     Status  Salary
0      101  Employee_101  Marketing   Contract   19000
1      102  Employee_102      Admin   Contract   19000
2      103  Employee_103    Finance  Permanent   23120
3      104  Employee_104      Admin  Permanent   21616
4      105  Employee_105      Admin  Permanent   29257
5      106  Employee_106         IT   Contract   19000
6      107  Employee_107    Finance   Contract   19000
7      108  Employee_108    Finance  Permanent   35445
8      109  Employee_109    Finance  Permanent   31646
9      110  Employee_110      Admin  Permanent   29075
```

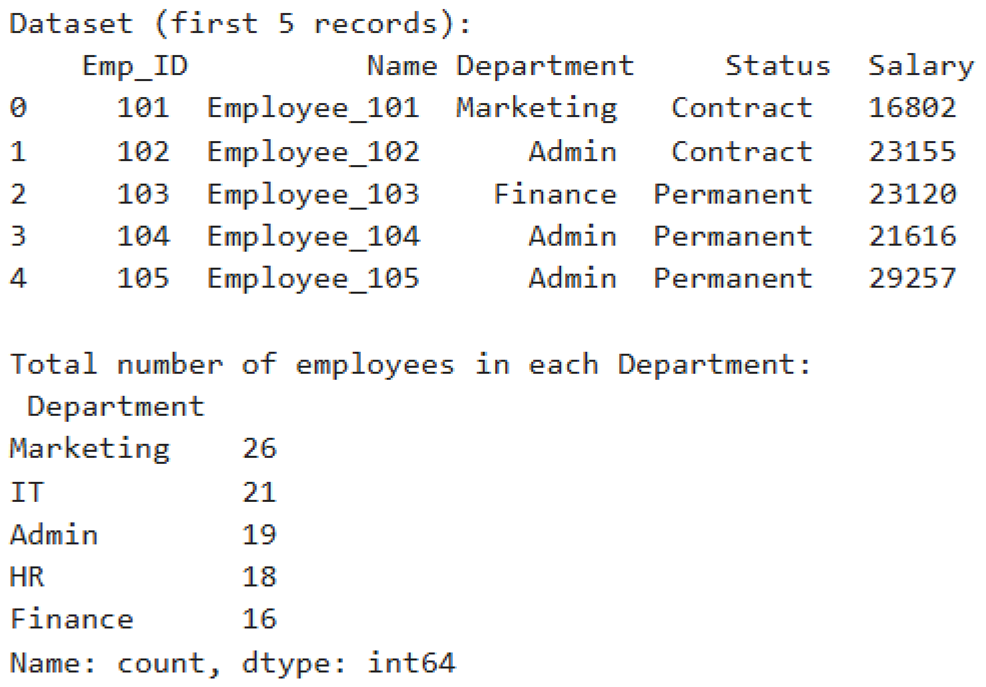
Practical No:10

Aim: Count the total number of employees in each Department.

Code:

```
import pandas as pd
# Step 1: Load the dataset
df = pd.read_csv("employee_data.csv")
# Step 2: Display first few records
print("Dataset (first 5 records):\n", df.head())
# Step 3: Count total number of employees in each Department
dept_counts = df['Department'].value_counts()
# Step 4: Display the counts
print("\nTotal number of employees in each Department:\n", dept_counts)
```

Output:



```
Dataset (first 5 records):
    Emp_ID          Name Department     Status  Salary
0      101  Employee_101  Marketing   Contract   16802
1      102  Employee_102      Admin   Contract   23155
2      103  Employee_103    Finance  Permanent   23120
3      104  Employee_104      Admin  Permanent   21616
4      105  Employee_105      Admin  Permanent   29257

Total number of employees in each Department:
 Department
Marketing    26
IT           21
Admin        19
HR           18
Finance      16
Name: count, dtype: int64
```

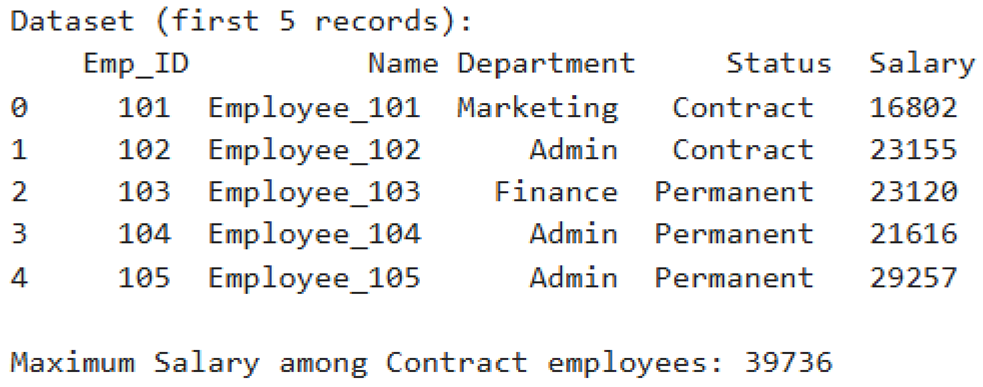
Practical No:11

Aim: Display the maximum salary among employees with ‘Contract’ status.

Code:

```
import pandas as pd
# Step 1: Load the dataset
df = pd.read_csv("employee_data.csv")
# Step 2: Display first few records
print("Dataset (first 5 records):\n", df.head())
# Step 3: Filter Contract employees and find maximum salary
max_contract_salary = df[df['Status'] == 'Contract']['Salary'].max()
# Step 4: Display the result
print("\nMaximum Salary among Contract employees:", max_contract_salary)
```

Output:



```
Dataset (first 5 records):
    Emp_ID          Name Department     Status  Salary
0     101  Employee_101  Marketing   Contract   16802
1     102  Employee_102      Admin   Contract   23155
2     103  Employee_103    Finance  Permanent   23120
3     104  Employee_104      Admin  Permanent   21616
4     105  Employee_105      Admin  Permanent   29257

Maximum Salary among Contract employees: 39736
```

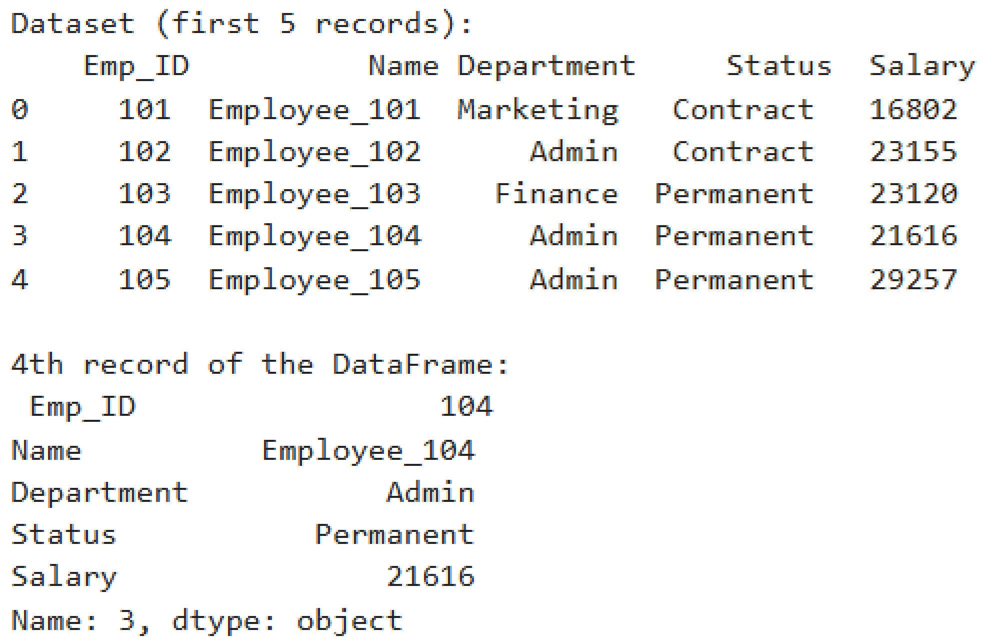
Practical No:12

Aim: Display the 4th record of the DataFrame.

Code:

```
import pandas as pd

# Step 1: Load the dataset
df = pd.read_csv("employee_data.csv")

# Step 2: Display first few records to verify
print("Dataset (first 5 records):\n", df.head())

# Step 3: Display the 4th record (index 3)
fourth_record = df.iloc[3]
print("\n4th record of the DataFrame:\n", fourth_record)
```

Output:



```
Dataset (first 5 records):
    Emp_ID          Name Department     Status  Salary
0     101  Employee_101  Marketing   Contract   16802
1     102  Employee_102      Admin   Contract   23155
2     103  Employee_103    Finance  Permanent   23120
3     104  Employee_104      Admin  Permanent   21616
4     105  Employee_105      Admin  Permanent   29257

4th record of the DataFrame:
 Emp_ID                 104
Name          Employee_104
Department           Admin
Status           Permanent
Salary               21616
Name: 3, dtype: object
```