Name – Neha R More

Roll No. – 745   Batch – G3

Practical No. 2

Problem statement:- Prepare/Take datasets for any real-life application.And perform operations on it. **CSV File:- examfees.csv**

PRN No	Name of Student	Registered Courses no.	Semester1	fees paid for reexam	category	admission date
2.02E+11	2.02E+11 1	1	Paid	500	OBC	20-Oct-22
2.02E+11		1	Paid	500	GENERAL	16-Nov22
2.02E+11	341	8	Paid	4000	OBC	20-Oct-22
2.02E+11	2.02E+11	1	Paid	500	OBC	20-Oct-22
2.02E+11	2.02E+11	1	#N/A	500	VJ/NT	20-Oct-22
2.02E+11	737	2	Paid	1000	GENERAL	20-Oct-22
2.02E+11	331	2	Paid	1000	VJ/NT	20-Oct-22
2.02E+11	305 Sahil Dipak Ingale	1	Paid	500	OBC	2-Nov-22
2.02E+11		3	Paid	1500	OBC	2-Nov-22
2.02E+11	Rajat Vijay Hattimare	2	Paid	1000	SBC	20-Oct-22
2.02E+11	Pratham Mishra	2	#N/A	1000	OBC	1-Nov-22
2.02E+11	306	6	Paid	3000	GENERAL	19-Oct-22
2.02E+11		3	Paid	1500	OBC	20-Oct-22
2.02E+11	Sudhanshu Nagraj Pendor					
2.02E+11	Shinde Yashvardhan sadashiv	2	Paid	1000	GENERAL	30-Nov-22
2.02E+11	Kahalkar pranay vasant	1	Paid	500	GENERAL	20-Oct-22
2.02E+11	VAISHNAVI EKLASPUR	2	Paid	1000	GENERAL	30-Nov-22
2.02E+11	Rushikesh Dilip Gawande	2	Paid	1000	OBC	20-Oct-22

2.02E+11	Yash sanjay dhunde	3	Paid	1500	OBC	19-Oct-22
2.02E+11	Durvesh jagtap	4	Paid	2000	OBC	16-Nov-22
2.02E+11	Shravan Vyankatesh Bobade	1	Paid	500	SC	20-Oct-22
2.02E+11	Rohit Basuraj Aheri	1	Paid	500	VJ/NT	20-Oct-22
2.02E+11	Vaishnavi Deepak Mirje	6	Paid	3000	OBC	21-Oct-22
2.02E+11	Nirmiti kishor raut	2	Paid	1000	SC	20-Oct-22
2.02E+11	Roshan yuvraj jadhav	3	Paid	1500	VJ/NT	26-Nov-22
2.02E+11	Rutvik Rajendra Pawar	1	Paid	500	OBC	16-Nov-

2.02E+11	Pradyumna Kishor Kulkarni	1	Paid	500	GENERAL	30-Nov-22
2.02E+11	Ritika Sanjay Choudhari	2	Paid	1000	GENERAL	12-Nov-22
2.02E+11	Vaishnavi Subhash Pawar	3	Paid	1500	VJ/NT	20-Oct-22
2.02E+11	Atharva Kishor Bramhankar	2	Paid	1000	VJ/NT	20-Oct-22

```python
registered_courses_no=[]
semister1=[]
fees_paid_reexam=dict()
category=[]
admission_date=[]

fp1=open("exam fees.csv","r")
data=fp1.readline()

while(True):


data=fp1.readline()
if not data:
break;
    #print(data)
    #data=data.replace("\n","")
temp=data.split(",")
registered_courses_no.append(temp[3])
semister1.append(temp[4])
fees_paid_reexam.update({temp[1]:temp[5]})
category.append(temp[6])
admission_date.append(temp[7])

fp1.close()

semister1=tuple(semister1)
print(type(semister1))
```

output-

```python
<class 'tuple'>
print("\nregistered_courses_no\n",registered_courses_no,end="")
print("\nsemister1\n",semister1,end="")
```

```python
print("\nfees_paid_reexam\n",fees_paid_reexam,end="")
print("\ncategory\n",category,end="")
print("\nadmission_date\n",admission_date,end="")
```

Output:-

registered_courses_no
 ['1', '1', '8', '1', '1', '2', '2', '1', '3', '2', '2', '6',
'3', '2', '1', '2', '2', '3', '4', '1', '1', '6', '2', '3', '1',
'1', '2', '3', '2'] semister1
 ('Paid', 'Paid', 'Paid', 'Paid', '#N/A', 'Paid', 'Paid',
'Paid', 'Paid', 'Paid', '#N/A', 'Paid', 'Paid', 'Paid', 'Paid',
'Paid', 'Paid', 'Paid', 'Paid', 'Paid', 'Paid', 'Paid', 'Paid',
'Paid', 'Paid', 'Paid', 'Paid', 'Paid', 'Paid') fees_paid_reexam
{'2.02201E+11': '500', '1': '500', '341': '4000', '737': '1000',
'331': '1000', '305': '500', 'Sahil Dipak Ingale': '1500',
'Rajat Vijay Hattimare ': '1000', 'Pratham Mishra': '1000',
'306': '3000', 'Sudhanshu Nagraj Pendor ': '1500', 'Shinde
Yashvardhan sadashiv ': '1000', 'Kahalkar pranay vasant': '500',
'VAISHNAVI EKLASPUR ': '1000', 'Rushikesh Dilip Gawande ':
'1000', 'Yash sanjay dhunde ': '1500', 'Durvesh jagtap': '2000',
'Shravan Vyankatesh Bobade ': '500', 'Rohit Basuraj Aheri ':
'500', 'Vaishnavi Deepak Mirje ': '3000', 'Nirmiti kishor raut':
'1000', 'Roshan yuvraj jadhav ': '1500', 'Rutvik Rajendra Pawar
': '500', 'Pradyumna Kishor Kulkarni ': '500', 'Ritika Sanjay
Choudhari ': '1000', 'Vaishnavi Subhash Pawar': '1500', 'Atharva
Kishor Bramhankar ': '1000'} category ['OBC', 'GENERAL', 'OBC',
'OBC', 'VJ/NT', 'GENERAL', 'VJ/NT', 'OBC', 'OBC', 'SBC', 'OBC',
'GENERAL', 'OBC', 'GENERAL', 'GENERAL', 'GENERAL', 'OBC', 'OBC',
'OBC', 'SC', 'VJ/NT', 'OBC', 'SC', 'VJ/NT', 'OBC', 'GENERAL',
'GENERAL', 'VJ/NT', 'VJ/NT'] admission_date

 ['20-Oct-22\n', '16-Nov-22\n', '20-Oct-22\n', '20-Oct-22\n',
'20-Oct-22\n', '20-Oct-22\n', '20-Oct-22\n', '2-Nov-22\n', '2-
Nov-22\n', '20-Oct-22\n', '1-Nov-22\n', '19-Oct-22\n', '20-Oct-
22\n', '30-Nov-22\n', '20-Oct-22\n', '30-Nov-22\n', '20-Oct-
22\n', '19-Oct-22\n', '16-Nov-22\n', '20-Oct-22\n', '20-Oct-
22\n', '21-Oct-22\n', '20-Oct-22\n', '26-Nov-22\n', '16-Nov-
22\n', '30-Nov-22\n', '12-Nov-22\n', '20-Oct-22\n', '20-Oct-
22\n']

1.

```python
frequency={}#{registered_courses_no}
#iteracting over the list for
item in admission_date:
  #checking  the element in dictionary
if item in frequency:
#incrementing the counter
frequency[item]+=1    else:
     #initialising the count        frequency[item]=1 #printing
the frequency print(frequency)
newlist=sorted(frequency.items(),key=lambda x:x[1],reverse=True)
sortdict=dict(newlist) print(sortdict) print("the date at which
most admissions taken
at",list(sortdict.keys())[0],"by",list(sortdict.values())[0],"st
udents")
```

Output:-

```
{'20-Oct-22\n': 15, '16-Nov-22\n': 3, '2-Nov-22\n': 2, '1-Nov-
22\n': 1, '19-Oct-22\n': 2, '30-Nov-22\n': 3, '21-Oct-22\n': 1,
'26-Nov-22\n': 1, '12-Nov-22\n': 1}
{'20-Oct-22\n': 15, '16-Nov-22\n': 3, '30-Nov-22\n': 3, '2-Nov-
22\n': 2, '19-Oct-22\n': 2, '1-Nov-22\n': 1, '21-Oct-22\n': 1,
'26-Nov-22\n': 1, '12-Nov-22\n': 1}
the date at which most admissions taken at 20-Oct-22 by
15 students
```

2.

```python
frequency={}#{registered_courses_no}
#iteracting over the list for item in
category: #checking the element in
dictionary if item in frequency:
#incrementing the counter
frequency[item]+=1 else:
     #initialising the count
frequency[item]=1 #printing the
frequency print(frequency)
```

```python
newlist=sorted(frequency.items(),key=lambda x:x[1],reverse=True)
sortdict=dict(newlist) print(sortdict) print("the most time
category found is
",list(sortdict.keys())[0],"in",list(sortdict.values())[0],"stud
ents")
```

Output:-

```
{'OBC': 12, 'GENERAL': 8, 'VJ/NT': 6, 'SBC': 1, 'SC': 2}
{'OBC': 12, 'GENERAL': 8, 'VJ/NT': 6, 'SC': 2, 'SBC': 1}
the most time category found is OBC in 12 students
```

3.

```python
frequency={}#{semister1}
#iteracting over the list
for item in semister1:
  #checking  the element in
dictionary    if item in frequency:
#incrementing the counter
frequency[item]+=1    else:
      #initialising the count         frequency[item]=1 #printing
the                    frequency                       print(frequency)
newlist=sorted(frequency.items(),key=lambda x:x[1],reverse=True)
sortdict=dict(newlist) print(sortdict) print("the fees
got",list(sortdict.keys())[0],"by",list(sortdict.values())
[0],"students")
```

Output:-

```
{'Paid': 27, '#N/A': 2} {'Paid':
27, '#N/A': 2} the fees  got Paid
by 27 students
```

4.

```
frequency={}#{registered_courses_no}
#iteracting over the list for item in
category: #checking the element in
dictionary if item in frequency:
#incrementing the counter
frequency[item]+=1 else:
      #initialising the count
frequency[item]=1 #printing the frequency
print(frequency)
newlist=sorted(frequency.items(),key=lambda
x:x[1]) sortdict=dict(newlist) print(sortdict)
print("the least time category found is
",list(sortdict.keys())[0],"in",list(sortdict.values())[0],"stud
ents")
```

Output:-

```
{'OBC': 12, 'GENERAL': 8, 'VJ/NT': 6, 'SBC': 1, 'SC': 2}
{'SBC': 1, 'SC': 2, 'VJ/NT': 6, 'GENERAL': 8, 'OBC': 12}
the least time category found is SBC in 1 students
```

5.

```
frequency={}#{semister1}
#iteracting over the list for
item in semister1:
  #checking the element in
dictionary if item in frequency:
#incrementing the counter
frequency[item]+=1 else:
      #initialising the count

      frequency[item]=1  #printing  the  frequency  print(frequency)
newlist=sorted(frequency.items(),key=lambda          x:x[1])
sortdict=dict(newlist) print(sortdict) print("the fees
got",list(sortdict.keys())[0],"by",list(sortdict.values())
[0],"students")
```

Output:-

```
{'Paid': 27, '#N/A': 2} {'#N/A': 2, 'Paid': 27} the fees got
#N/A by 2 students
```

6.

```
frequency={}#{registered_courses_no}
#iteracting over the list for item in
registered_courses_no: #checking the
element in dictionary if item
in frequency:      #incrementing the
counter      frequency[item]+=1
else:
      #initialising the count
frequency[item]=1 #printing the frequency
print(frequency)
newlist=sorted(frequency.items(),key=lambda
x:x[1]) sortdict=dict(newlist) print(sortdict)


print("the most course no. registered
is",list(sortdict.keys())[0],"by",list(sortdict.values())[0],"st
udents")
```

Output:-

```
{'1': 10, '8': 1, '2': 10, '3': 5, '6': 2, '4': 1}
{'8': 1, '4': 1, '6': 2, '3': 5, '1': 10, '2': 10} the
most course no. registered is 8 by 1 students
```

7.

```
frequency={}#{registered_courses_no}
#iteracting over the list for
item in admission_date:
  #checking  the element in
dictionary    if item in frequency:
#incrementing the counter
frequency[item]+=1    else:
     #initialising the count         frequency[item]=1 #printing
the frequency print(frequency)
newlist=sorted(frequency.items(),key=lambda x:x[1])
sortdict=dict(newlist) print(sortdict) print("the date at at
which less admissions taken
is",list(sortdict.keys())[0],"by",list(sortdict.values())[0],"st
udents")
```

Output:-

```
{'20-Oct-22\n': 15, '16-Nov-22\n': 3, '2-Nov-22\n': 2, '1-Nov-
22\n': 1, '19-Oct-22\n': 2, '30-Nov-22\n': 3, '21-Oct-22\n': 1,
'26-Nov-22\n': 1, '12-Nov-22\n': 1}
{'1-Nov-22\n': 1, '21-Oct-22\n': 1, '26-Nov-22\n': 1, '12-
Nov22\n': 1, '2-Nov-22\n': 2, '19-Oct-22\n': 2, '16-Nov-22\n': 3,
'30-Nov-22\n': 3, '20-Oct-22\n': 15}
the date at at which less admissions taken is 1-Nov-22 by
1 students
```

8.

```
frequency={}#{registered_courses_no}
#iteracting over the list for item in
fees_paid_reexam.values():
#checking the element in dictionary
if item in frequency: #incrementing
the counter frequency[item]+=1
else: #initialising the count the
frequency print(frequency)          frequency[item]=1 #printing

newlist=sorted(frequency.items(),key=lambda x:x[1])
sortdict=dict(newlist) print(sortdict) print("the most amount
paid
is",list(sortdict.keys())[0],"by",list(sortdict.values())[0],"st
udents")
```

Output:-

```
{'500': 8, '4000': 1, '1000': 10, '1500': 5, '3000': 2, '2000':
1}
```

9.

```
frequency={}#{registered_courses_no}
#iteracting over the list for item in
fees_paid_reexam.values(): #checking
the element in dictionary if item in
frequency:
```

```
    #incrementing the
counter
frequency[item]+=1    else:
       #initialising the count         frequency[item]=1 #printing
the frequency print(frequency)
newlist=sorted(frequency.items(),key=lambda x:x[1],reverse=True)
sortdict=dict(newlist) print(sortdict) print("the most times
amount paid
is",list(sortdict.keys())[0],"by",list(sortdict.values())[0],"st
udents")
```

Output:-

```
{'500': 8, '4000': 1, '1000': 10, '1500': 5, '3000': 2, '2000':
1} {'1000': 10, '500': 8, '1500': 5, '3000': 2, '4000': 1,
'2000':
1}
the most times amount paid is 1000 by 10 students
```

10.

```
frequency={}#{registered_courses_no}
#iteracting over the list for item in
fees_paid_reexam.values():
#checking  the element in dictionary
if item in frequency:
#incrementing the counter
frequency[item]+=1    else:
      #initialising the count          frequency[item]=1 #printing
the frequency print(frequency)
newlist=sorted(frequency.items(),key=lambda x:x[1])
sortdict=dict(newlist) print(sortdict) print("the most amount
paid
is",list(sortdict.keys())[0],"by",list(sortdict.values())[0],"st
udents")
```

Output:-

```
{'500': 8, '4000': 1, '1000': 10, '1500': 5, '3000': 2, '2000':
1}
```