In [ ]:

```
import numpy as np
employee_table = np.array([
    [1000, 'Torbati', 'Yolanda', 'F', 'Programmer'],
    [1001, 'Kleinn', 'Joel', 'M', 'Programmer'],
    [1002, 'Ginsburg', 'Laura', 'F', 'President'],
    [1003, 'Cox', 'Jennifer', 'F', 'Programmer'],
    [1005, 'Ziada', 'Mauri', 'M', 'Product Designer'],
    [1006, 'Keyser', 'Cara', 'F', 'Account Executive'],
    [1010, 'Smith', 'Roxie', 'M', 'Programmer'],
    [1011, 'Nelson', 'Robert', 'M', 'Programmer'],
    [1012, 'Sachsen', 'Lars', 'M', 'Support Technician'],
    [1013, 'Shannon', 'Don', 'M', 'Product Designer']
])
print(employee_table)
```

```
[['1000' 'Torbati' 'Yolanda' 'F' 'Programmer']
 ['1001' 'Kleinn' 'Joel' 'M' 'Programmer']
 ['1002' 'Ginsburg' 'Laura' 'F' 'President']
 ['1003' 'Cox' 'Jennifer' 'F' 'Programmer']
 ['1005' 'Ziada' 'Mauri' 'M' 'Product Designer']
 ['1006' 'Keyser' 'Cara' 'F' 'Account Executive']
 ['1010' 'Smith' 'Roxie' 'M' 'Programmer']
 ['1011' 'Nelson' 'Robert' 'M' 'Programmer']
 ['1012' 'Sachsen' 'Lars' 'M' 'Support Technician']
 ['1013' 'Shannon' 'Don' 'M' 'Product Designer']]
```

In [ ]:

```
#How many Male employees are in a company
a=np.sum(employee_table[:,3]=='M')
print("Number of Male Employees in the Company:", a)
```

```
Number of Male Employees in the Company: 6
```

In [ ]:

```
#Display the details of employees whose Last_Name starts with S
b = [row for row in employee_table if row[1].startswith('S')]

for employee in b:
    print("Employee ID:", employee[0])
    print("Last Name:", employee[1])
    print("First Name:", employee[2])
    print("Gender:", employee[3])
    print("Position:", employee[4])
    print()
```

```
Employee ID: 1010
Last Name: Smith
First Name: Roxie
Gender: M
Position: Programmer

Employee ID: 1012
Last Name: Sachsen
First Name: Lars
Gender: M
Position: Support Technician

Employee ID: 1013
Last Name: Shannon
First Name: Don
Gender: M
Position: Product Designer
```

In [ ]:

```
#Sort the Female Employee details in descending order based on First_Name

female= employee_table[employee_table[:,3]=='F']
print(female)
sort_female=female[np.argsort(female[:, 2])[::-1]]
for employee in sort_female:
    print("Employee ID:", employee[0])
    print("Last Name:", employee[1])
    print("First Name:", employee[2])
    print("Gender:", employee[3])
    print("Position:", employee[4])
    print()
```

```
[['1000' 'Torbati' 'Yolanda' 'F' 'Programmer']
 ['1002' 'Ginsburg' 'Laura' 'F' 'President']
 ['1003' 'Cox' 'Jennifer' 'F' 'Programmer']
 ['1006' 'Keyser' 'Cara' 'F' 'Account Executive']]
Employee ID: 1000
Last Name: Torbati
First Name: Yolanda
Gender: F
Position: Programmer

Employee ID: 1002
Last Name: Ginsburg
First Name: Laura
Gender: F
Position: President

Employee ID: 1003
Last Name: Cox
First Name: Jennifer
Gender: F
Position: Programmer

Employee ID: 1006
Last Name: Keyser
First Name: Cara
Gender: F
Position: Account Executive
```

In [ ]:

```python
#Extract 1D array and reshape it into 2D array.

employee1d = employee_table[:, 0]


employee2d = employee1d.reshape(-1, 1)


print("Original 1D array of Employee IDs:")
print(employee1d)


print("\nReshaped 2D array of Employee IDs:")
print(employee2d)
```

```
Original 1D array of Employee IDs:
['1000' '1001' '1002' '1003' '1005' '1006' '1010' '1011' '1012' '1013']

Reshaped 2D array of Employee IDs:
[['1000']
 ['1001']
 ['1002']
 ['1003']
 ['1005']
 ['1006']
 ['1010']
 ['1011']
 ['1012']
 ['1013']]
```

In [ ]:

```python
female_mask = (employee_table[:, 3] == 'F')
female_employee_matrix = employee_table[female_mask][:, [0, 1, 4]]
print("Extracted Matrix for Female Employees:")
print(female_employee_matrix)
```

```
Extracted Matrix for Female Employees:
[['1000' 'Torbati' 'Programmer']
 ['1002' 'Ginsburg' 'President']
 ['1003' 'Cox' 'Programmer']
 ['1006' 'Keyser' 'Account Executive']]
```

In [ ]:

In [ ]:

```python
import pandas as pd
r=pd.read_csv("D:/christ/python/lab7/bank.csv")
print(r)
print("\n\n")
print("Describing The table\n\n",r.describe())
print("\n\n")
e,f=r.shape
print("Shape of the CSV file\n\n",r.shape)
print("Number of Rows:",e,"\nNumber of Columns:",f,"\n")
print(r.info())
```

```
     age           job   marital  education  balance housing loan  day month
0     59        admin.   married  secondary     2343     yes   no    5   may  \
1     56        admin.   married  secondary       45      no   no    5   may
2     41    technician   married  secondary     1270     yes   no    5   may
3     55      services   married  secondary     2476     yes   no    5   may
4     54        admin.   married   tertiary      184      no   no    5   may
..   ...           ...       ...        ...      ...     ...  ...  ...   ...
194   26    management    single   tertiary     1004     yes   no   30   may
195   45    technician    single  secondary      410     yes   no   30   may
196   31    unemployed  divorced    primary        0     yes   no   30   may
197   29    management    single   tertiary        0     yes   no   30   may
198   53  entrepreneur    single  secondary       62     yes   no   30   may

     duration  campaign
0      1042.0         1
1      1467.0         1
2      1389.0         1
3       579.0         1
4       673.0         2
..        ...       ...
194     228.0         2
195     891.0         4
196     745.0        12
197     539.0        12
198    1044.0         5

[199 rows x 11 columns]



Describing The table

              age       balance         day     duration    campaign
count  199.000000    199.000000  199.000000   194.000000  199.000000
mean    38.492462   1282.366834   19.180905   919.927835    2.361809
std      9.328391   2099.805504    7.776385   380.936581    2.042416
min     23.000000   -538.000000    5.000000   182.000000    1.000000
25%     31.000000    101.000000   14.000000   653.250000    1.000000
50%     36.000000    541.000000   20.000000   854.500000    2.000000
75%     44.000000   1394.500000   27.000000  1100.750000    3.000000
max     60.000000  14481.000000   30.000000  3094.000000   15.000000



Shape of the CSV file

 (199, 11)
Number of Rows: 199
Number of Columns: 11

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 199 entries, 0 to 198
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        199 non-null    int64
 1   job        199 non-null    object 
 2   marital    199 non-null    object 
 3   education  199 non-null    object 
 4   balance    199 non-null    int64  
 5   housing    199 non-null    object 
 6   loan       199 non-null    object 
 7   day        199 non-null    int64  
 8   month      199 non-null    object 
 9   duration   194 non-null    float64
 10  campaign   199 non-null    int64  
dtypes: float64(1), int64(4), object(6)
memory usage: 17.2+ KB
None
```

In [ ]:

```
r.fillna(method ='bfill')
```

Out[ ]:

	age	job	marital	education	balance	housing	loan	day	month	duratio
0	59	admin.	married	secondary	2343	yes	no	5	may	1042.
1	56	admin.	married	secondary	45	no	no	5	may	1467.
2	41	technician	married	secondary	1270	yes	no	5	may	1389.
3	55	services	married	secondary	2476	yes	no	5	may	579.
4	54	admin.	married	tertiary	184	no	no	5	may	673.
...	...	...	...	...	...	...	...	...	...	.
194	26	management	single	tertiary	1004	yes	no	30	may	228.
195	45	technician	single	secondary	410	yes	no	30	may	891.
196	31	unemployed	divorced	primary	0	yes	no	30	may	745.
197	29	management	single	tertiary	0	yes	no	30	may	539.
198	53	entrepreneur	single	secondary	62	yes	no	30	may	1044.

199 rows × 11 columns

In [ ]:

```
r.fillna(method ='ffill')
```

Out[ ]:

	age	job	marital	education	balance	housing	loan	day	month	duratio
0	59	admin.	married	secondary	2343	yes	no	5	may	1042.
1	56	admin.	married	secondary	45	no	no	5	may	1467.
2	41	technician	married	secondary	1270	yes	no	5	may	1389.
3	55	services	married	secondary	2476	yes	no	5	may	579.
4	54	admin.	married	tertiary	184	no	no	5	may	673.
...	...	...	...	...	...	...	...	...	...	.
194	26	management	single	tertiary	1004	yes	no	30	may	228.
195	45	technician	single	secondary	410	yes	no	30	may	891.
196	31	unemployed	divorced	primary	0	yes	no	30	may	745.
197	29	management	single	tertiary	0	yes	no	30	may	539.
198	53	entrepreneur	single	secondary	62	yes	no	30	may	1044.

199 rows × 11 columns

In [ ]:

```
r.dropna()
```

Out[ ]:

	age	job	marital	education	balance	housing	loan	day	month	duratio
0	59	admin.	married	secondary	2343	yes	no	5	may	1042.
1	56	admin.	married	secondary	45	no	no	5	may	1467.
2	41	technician	married	secondary	1270	yes	no	5	may	1389.
3	55	services	married	secondary	2476	yes	no	5	may	579.
4	54	admin.	married	tertiary	184	no	no	5	may	673.
...	...	...	...	...	...	...	...	...	...	.
194	26	management	single	tertiary	1004	yes	no	30	may	228.
195	45	technician	single	secondary	410	yes	no	30	may	891.
196	31	unemployed	divorced	primary	0	yes	no	30	may	745.
197	29	management	single	tertiary	0	yes	no	30	may	539.
198	53	entrepreneur	single	secondary	62	yes	no	30	may	1044.

194 rows × 11 columns

In [ ]:

```
d=r.groupby('marital')
```

```
d.first()
```

Out[ ]:

	age	job	education	balance	housing	loan	day	month	duration	ca
marital										
divorced	60	retired	secondary	545	yes	no	6	may	1030.0	
married	59	admin.	secondary	2343	yes	no	5	may	1042.0	
single	42	management	tertiary	0	yes	yes	5	may	562.0	