In [1]:

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

In [2]:

```
df = pd.DataFrame({'From_To': ['LoNDon_paris', 'MAdrid_miLAN',
'londON_StockhOlm',

'Budapest_PaRis', 'Brussels_londOn'],
'FlightNumber': [10045, np.nan, 10065, np.nan, 10085],
'RecentDelays': [[23, 47], [], [24, 43, 87], [13], [67, 32]],
'Airline': ['KLM(!)', '<Air France> (12)', '(British Airways. )',
'12. Air France', '"Swiss Air"']})
```

In [3]:

```
df
```

Out[3]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	NaN	[]	<Air France> (12)
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	NaN	[13]	12. Air France
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [4]:

```
#1. Some values in the the FlightNumber column are missing. These numbers are
#meant to increase by 10 with each row so 10055 and 10075 need to be put in
#place. Fill in these missing numbers and make the column an integer column
#(instead of a float column).
```

In [5]:

```
df['FlightNumber'].loc[np.where(df['FlightNumber'] .isnull())] = [df['FlightNumber'][x-1]+1
```

```
C:\Users\subha\anaconda3\lib\site-packages\pandas\core\indexing.py:670: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  self._setitem_with_indexer(indexer, value)
```

In [6]:

```
df['FlightNumber']
```

Out[6]:

```
0    10045.0
1    10055.0
2    10065.0
3    10075.0
4    10085.0
Name: FlightNumber, dtype: float64
```

In [7]:

```
#2. The From_To column would be better as two separate columns! Split each
#string on the underscore delimiter _ to give a new temporary DataFrame with
#the correct values. Assign the correct column names to this temporary
#DataFrame.
#3. Notice how the capitalisation of the city names is all mixed up in this
#temporary DataFrame. Standardise the strings so that only the first letter is
#uppercase (e.g. "londON" should become "London".)
#4. Delete the From_To column from df and attach the temporary DataFrame
#from the previous questions.
```

In [8]:

```
fromto = df.From_To.str.split('_',expand = True)
```

In [9]:

```
fromto
```

Out[9]:

	0	1
0	LoNDon	paris
1	MAdrid	miLAN
2	londON	StockhOlm
3	Budapest	PaRis
4	Brussels	londOn

In [10]:

```
departure = fromto[0]
```

In [11]:

```
arrival = fromto[1]
```

In [12]:

```
df['Departure'] = departure.str.title()
```

In [13]:

```
df['Arrival']=arrival.str.title()
```

In [14]:

```
df.drop('From_To',axis = 1,inplace = True)
```

In [15]:

```
df = df[['Departure','Arrival','Airline','FlightNumber','RecentDelays']]
df
```

Out[15]:

	Departure	Arrival	Airline	FlightNumber	RecentDelays
0	London	Paris	KLM(!)	10045.0	[23, 47]
1	Madrid	Milan	<Air France> (12)	10055.0	[]
2	London	Stockholm	(British Airways.)	10065.0	[24, 43, 87]
3	Budapest	Paris	12. Air France	10075.0	[13]
4	Brussels	London	"Swiss Air"	10085.0	[67, 32]

In [16]:

```
#5. In the RecentDelays column, the values have been entered into the
#DataFrame as a list. We would like each first value in its own column, each

#second value in its own column, and so on. If there isn't an Nth value, the value
#should be NaN.
#Expand the Series of lists into a DataFrame named delays, rename the columns
#delay_1, delay_2, etc. and replace the unwanted RecentDelays column in df
#with delays.
```

In [17]:

```
df
```

Out[17]:

	Departure	Arrival	Airline	FlightNumber	RecentDelays
0	London	Paris	KLM(!)	10045.0	[23, 47]
1	Madrid	Milan	<Air France> (12)	10055.0	[]
2	London	Stockholm	(British Airways.)	10065.0	[24, 43, 87]
3	Budapest	Paris	12. Air France	10075.0	[13]
4	Brussels	London	"Swiss Air"	10085.0	[67, 32]

In [18]:

```
#cleaning AIRLINECOLUMN.
```

In [19]:

```
strip  = df['Airline']
```

In [20]:

```
strip
```

Out[20]:

```
0                 KLM(!)
1      <Air France> (12)
2    (British Airways. )
3         12. Air France
4            "Swiss Air"
Name: Airline, dtype: object
```

In [21]:

```
df['Airline']=strip.str.strip('(').str.strip(')').str.strip('(').str.strip('<').str.strip('
```

In [22]:

```
df
```

Out[22]:

	Departure	Arrival	Airline	FlightNumber	RecentDelays
0	London	Paris	KLM	10045.0	[23, 47]
1	Madrid	Milan	Air France	10055.0	[]
2	London	Stockholm	British Airways	10065.0	[24, 43, 87]
3	Budapest	Paris	Air France	10075.0	[13]
4	Brussels	London	Swiss Air	10085.0	[67, 32]

In [23]:

```
#cleaning recent delays column
```

In [24]:

```
df1 = pd.DataFrame(df['RecentDelays'].tolist())
```

In [25]:

```
df1
```

Out[25]:

	0	1	2
0	23.0	47.0	NaN
1	NaN	NaN	NaN
2	24.0	43.0	87.0
3	13.0	NaN	NaN
4	67.0	32.0	NaN

In [26]:

```
df[["Delays_1","Delays_2","Delays_3"]] =  df1[[0,1,2]]
```

In [27]:

```
df.drop("RecentDelays",axis = 1,inplace =True)
```

In [28]:

```
df
```

Out[28]:

	Departure	Arrival	Airline	FlightNumber	Delays_1	Delays_2	Delays_3
0	London	Paris	KLM	10045.0	23.0	47.0	NaN
1	Madrid	Milan	Air France	10055.0	NaN	NaN	NaN
2	London	Stockholm	British Airways	10065.0	24.0	43.0	87.0
3	Budapest	Paris	Air France	10075.0	13.0	NaN	NaN
4	Brussels	London	Swiss Air	10085.0	67.0	32.0	NaN