In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
#Reading the required data file
data_frame = pd.read_csv('movies.dat',names = ['Movieid','Title','Genre'] ,  delimiter="::")
```

```
/var/folders/9z/75yp7_695hg75lbf0tpmjs9w0000gn/T/ipykernel_17320/3745429871.py:2: ParserWarning: Falling back to the 'python' engine because the 'c' engine does not support regex separators (separators > 1 char and different from '\s+' are interpreted as regex); you can avoid this warning by specifying engine='python'.
  data_frame = pd.read_csv('movies.dat',names = ['Movieid','Title','Genre'] ,  delimiter="::")
```

In [3]:
```
data_frame
```

Out[3]:

	Movieid	Title	Genre
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
10676	65088	Bedtime Stories (2008)	Adventure|Children|Comedy
10677	65091	Manhattan Melodrama (1934)	Crime|Drama|Romance
10678	65126	Choke (2008)	Comedy|Drama
10679	65130	Revolutionary Road (2008)	Drama|Romance
10680	65133	Blackadder Back & Forth (1999)	Comedy

10681 rows × 3 columns

In [4]:
```
data_frame2 = pd.read_csv('ratings.dat',names = ['Userid','Movieid','Rating','Timestamp'] ,  delimiter="::")
```

```
/var/folders/9z/75yp7_695hg75lbf0tpmjs9w0000gn/T/ipykernel_17320/919639521.py:1: ParserWarning: Falling back to the 'python' engine because the 'c' engine does not support regex separators (separators > 1 char and different from '\s+' are interpreted as regex); you can avoid this warning by specifying engine='python'.
  data_frame2 = pd.read_csv('ratings.dat',names = ['Userid','Movieid','Rating','Timestamp'] ,  delimiter="::")
```

In [5]:
```
data_frame2
```

Out[5]:

	Userid	Movieid	Rating	Timestamp
0	1	122	5.0	838985046
1	1	185	5.0	838983525
2	1	231	5.0	838983392
3	1	292	5.0	838983421
4	1	316	5.0	838983392
...	...	...	...	...
10000049	71567	2107	1.0	912580553
10000050	71567	2126	2.0	912649143
10000051	71567	2294	5.0	912577968
10000052	71567	2338	2.0	912578016
10000053	71567	2384	2.0	912578173

10000054 rows × 4 columns

In [6]:
```
#Merging the datasets
merge_frame = data_frame.merge(data_frame2, on='Movieid') #movie-id is the common attribute
```

In [7]:
```
merge_frame
```

Out[7]:

	Movieid	Title	Genre	Userid	Rating	Timestamp
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	5	1.0	857911264
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	14	3.0	1133572007
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	18	3.0	1111545931
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	23	5.0	849543482
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	24	5.0	868254237
...	...	...	...	...	...	...
10000049	65133	Blackadder Back & Forth (1999)	Comedy	24495	4.0	1231081348
10000050	65133	Blackadder Back & Forth (1999)	Comedy	33384	3.0	1231034528
10000051	65133	Blackadder Back & Forth (1999)	Comedy	40570	2.0	1231055397
10000052	65133	Blackadder Back & Forth (1999)	Comedy	45430	2.5	1231105425
10000053	65133	Blackadder Back & Forth (1999)	Comedy	68151	5.0	1231129793

10000054 rows × 6 columns

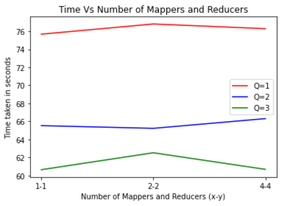
In [26]:
```
#Removing comma from values in column
merge_frame["Title"]=merge_frame["Title"].str.replace(',','.')
```

In [ ]:
```
#Converting to csv file
merge_frame.to_csv('merge_data5.csv', index=False )
```

Graph for plotting Time VS Number of Mappers and Reducers

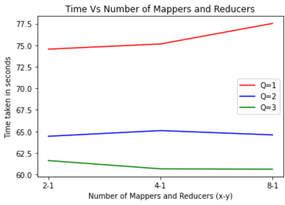
While changing both mappers and reducers for question 1, 2 and 3

In [33]:
```
time1=[75.68,76.81,76.29]
time2=[65.53,65.23,66.31]
time3=[60.64,62.52,60.68]
plt.title('Time Vs Number of Mappers and Reducers')
plt.xlabel('Number of Mappers and Reducers (x-y)')
plt.ylabel('Time taken in seconds')
plt.plot(["1-1","2-2","4-4"], time1, color='red',label="Q=1")
plt.plot(["1-1","2-2","4-4"], time2, color='blue',label="Q=2")
plt.plot(["1-1","2-2","4-4"], time3, color='green',label="Q=3")

plt.legend()
plt.show()
```



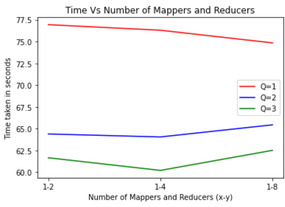
While changing only mappers and reducer=1 for question 1, 2 and 3

In [34]:
```
time1=[74.55,75.15,77.53]
time2=[64.44,65.10,64.60]
time3=[61.62,60.66,60.62]
plt.title('Time Vs Number of Mappers and Reducers')
plt.xlabel('Number of Mappers and Reducers (x-y)')
plt.ylabel('Time taken in seconds')
plt.plot(["2-1","4-1","8-1"], time1, color='red',label="Q=1")
plt.plot(["2-1","4-1","8-1"], time2, color='blue',label="Q=2")
plt.plot(["2-1","4-1","8-1"], time3, color='green',label="Q=3")

plt.legend()
plt.show()
```



While changing only reducers and mapper=1 for question 1, 2 and 3

In [32]:
```
time1=[76.94,76.30,74.85]
time2=[64.39,64.05,65.44]
time3=[61.66,60.21,62.51]
plt.title('Time Vs Number of Mappers and Reducers')
plt.xlabel('Number of Mappers and Reducers (x-y)')
plt.ylabel('Time taken in seconds')
plt.plot(["1-2","1-4","1-8"], time1, color='red',label="Q=1")
plt.plot(["1-2","1-4","1-8"], time2, color='blue',label="Q=2")
plt.plot(["1-2","1-4","1-8"], time3, color='green',label="Q=3")

plt.legend()
plt.show()
```



In [ ]:
```
```