In [1]:

```python
import pandas as pd
import os

# Initialize an empty DataFrame
all_months_data = pd.DataFrame()

# List all files in the directory
files = [file for file in os.listdir('D:/Sales_Data') if file.endswith('.csv')]

# Loop through each file and concatenate the DataFrames
for file in files:
    file_path = os.path.join('D:/Sales_Data', file)  # Construct the file path
    df = pd.read_csv(file_path)  # Read the CSV file into a DataFrame
    all_months_data = pd.concat([all_months_data, df])  # Concatenate the DataFrames

# Display the first few rows of the combined DataFrame
all_months_data.to_csv("all_data.csv",index=False)
```

In [3]:

```python
import pandas as pd
all_data=pd.read_csv("all_data.csv")
all_data.head()
```

Out[3]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	NaN	NaN	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

clean up data

In [5]:

```python
nan_df = all_data[all_data.isna().any(axis=1)]
nan_df.head()

all_data = all_data.dropna(how='all')
all_data.head()
```

Out[5]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001

In [6]:

```python
all_data = all_data[all_data['Order Date'].str[0:2] != 'Or']
```

convert nonint string into correct datatype

In [8]:

```python
all_data['Quantity Ordered']=pd.to_numeric(all_data['Quantity Ordered']) #maked int
all_data['Price Each']=pd.to_numeric(all_data['Price Each']) #maked float
```

In [ ]:

```python

```

!add a month column

In [10]:

```python
all_data['month'] = all_data['Order Date'].str[0:2]
all_data['month'] = all_data['month'].astype('int32')
all_data.head()
```

Out[10]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4

add sales column

In [12]:

```
all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
all_data.head()
```

Out[12]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Sales
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99

add city

In [14]:

```
def get_city(address):
    return address.split(',')[1]

def get_state(adress):
    return adress.split(',')[2].split(' ')[1]

all_data['City'] = all_data['Purchase Address'].apply(lambda x: get_city(x) +'('+ get_state(x)+')')
all_data.head()
```

Out[14]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas(TX)
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston(MA)
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles(CA)
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)

Q.1 what was the best months for the sales? how much was earned that month?

In [16]:

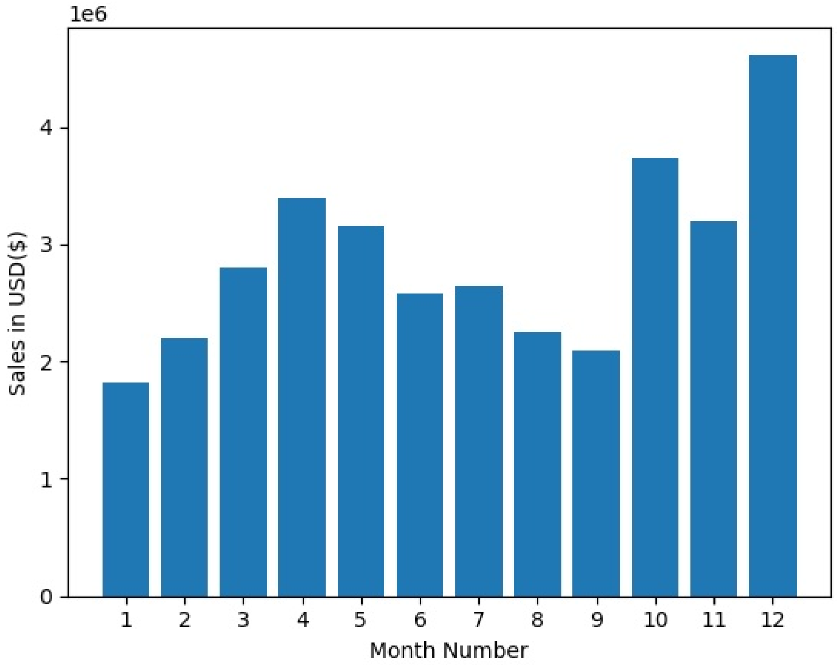
```
 results = all_data.groupby('month').sum()
```

In [17]:

```
import matplotlib.pyplot as plt

months = range(1,13)

plt.bar(months, results["Sales"])
plt.xticks(months)
plt.ylabel('Sales in USD($)')
plt.xlabel('Month Number')
plt.show()
```



Q.2 which city is a higher number of sales?

In [19]:

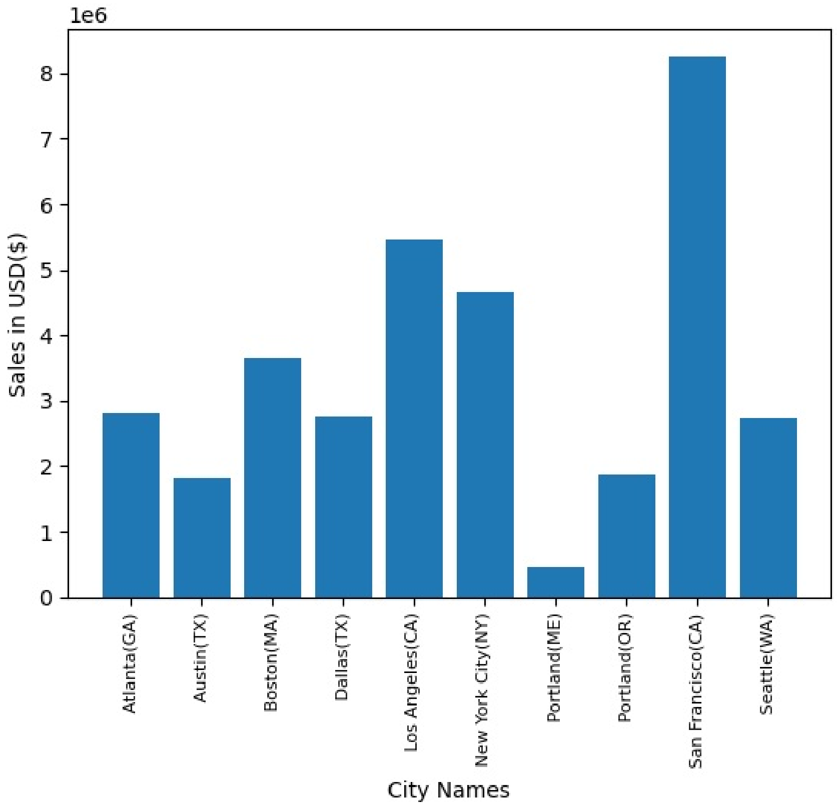
```
results = all_data.groupby('City').sum()
```

In [20]:

```
# import matplotlib.pyplot as plt

cities = [ City for City,df in all_data.groupby('City')]

plt.bar(cities, results['Sales'])
plt.xticks(cities,rotation='vertical',size=8)
plt.ylabel("Sales in USD($)")
plt.xlabel("City Names")
plt.show()
```



Q.3what time shoud we display advertisment to maximise

liklihood of customers buying product?

In [22]:
```
import datetime
all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_10768\3141956513.py:2: UserWarning: Could not infer format, so each
element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected
, please specify a format.
  all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

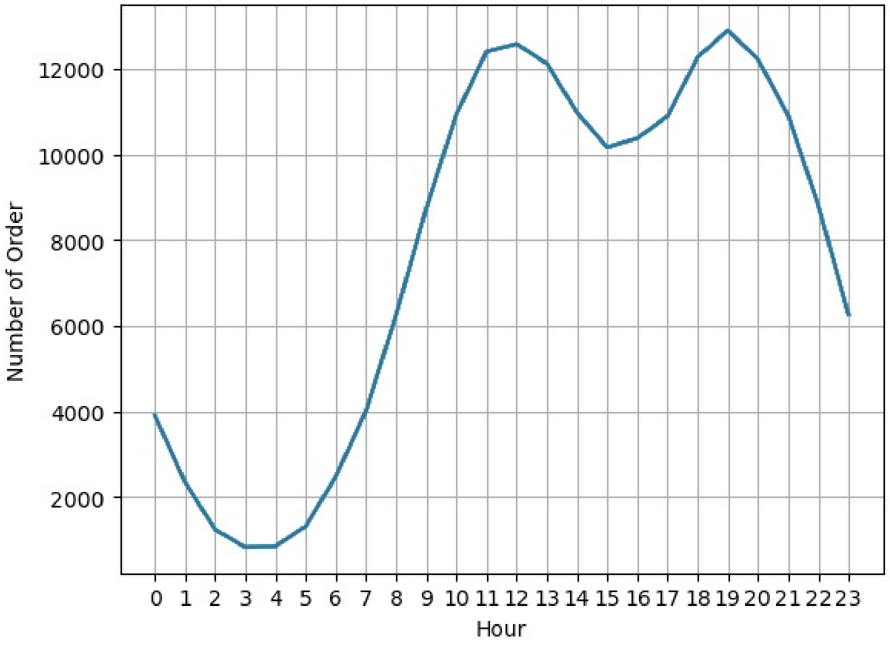
In [23]:
```
all_data['Hour'] = all_data['Order Date'].dt.hour
all_data['Minute'] = all_data['Order Date'].dt.minute
all_data['Count'] = 1
all_data.head()
```

Out[23]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Sales	City	Hour	Minute	Count
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas(TX)	8	46	1
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston(MA)	22	30	1
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles(CA)	14	38	1
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)	14	38	1
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)	9	27	1

In [24]:
```
hours = [hours for hours, df in all_data.groupby('Hour')]

plt.plot(hours, all_data.groupby(['Hour']).count())
plt.xticks(hours)
plt.xlabel('Hour')
plt.ylabel('Number of Order')
plt.grid()
plt.show()
```



Q.4 what product are most often sold togather?

In [26]:
```
df = all_data[all_data['Order ID'].duplicated(keep=False)]

df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
df = df[['Order ID','Grouped']].drop_duplicates()
df.head()
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_10768\2612133963.py:3: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retu
rning-a-view-versus-a-copy
  df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
```

Out[26]:

	Order ID	Grouped
3	176560	Google Phone,Wired Headphones
18	176574	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones,Bose SoundSport Hea...
32	176586	AAA Batteries (4-pack),Google Phone
119	176672	Lightning Charging Cable,USB-C Charging Cable

In [27]:

```python
import collections
from itertools import combinations
from collections import Counter

count = Counter()

for row in df['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 2)))

for key,value in count.most_common(10):
    print(key,value)
```

```
('iPhone', 'Lightning Charging Cable') 1005
('Google Phone', 'USB-C Charging Cable') 987
('iPhone', 'Wired Headphones') 447
('Google Phone', 'Wired Headphones') 414
('Vareebadd Phone', 'USB-C Charging Cable') 361
('iPhone', 'Apple Airpods Headphones') 360
('Google Phone', 'Bose SoundSport Headphones') 220
('USB-C Charging Cable', 'Wired Headphones') 160
('Vareebadd Phone', 'Wired Headphones') 143
('Lightning Charging Cable', 'Wired Headphones') 92
```

In [28]:

```python
all_data.head()
```

Out[28]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Sales	City	Hour	Minute	Count
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas(TX)	8	46	1
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston(MA)	22	30	1
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles(CA)	14	38	1
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)	14	38	1
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)	9	27	1

Q.5 What product sold the most? Why do u thnk it sold the most?

In [30]:

```python
all_data.head()
```

Out[30]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	Sales	City	Hour	Minute	Count
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas(TX)	8	46	1
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston(MA)	22	30	1
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles(CA)	14	38	1
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)	14	38	1
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles(CA)	9	27	1

In [31]:

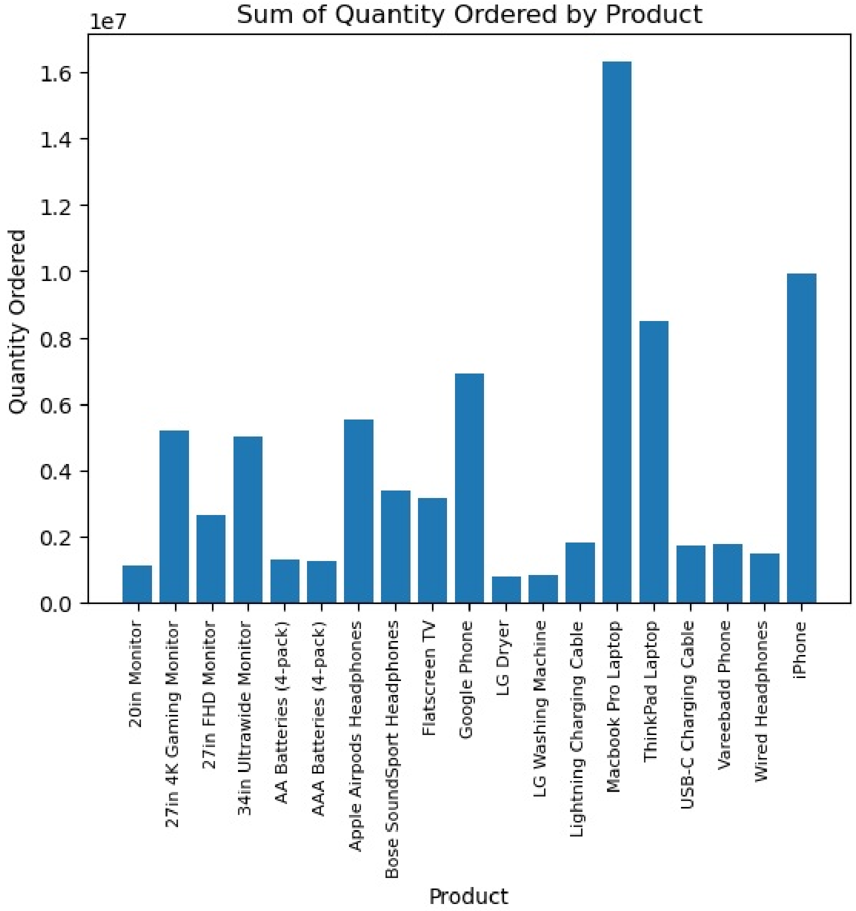
```
numeric_cols = [col for col in all_data.columns if all_data[col].dtype.kind in 'bifc']
product_group = all_data.groupby('Product')[numeric_cols].sum()
```

In [32]:

```
numeric_cols = [col for col in all_data.columns if all_data[col].dtype.kind in 'bifc']
product_group = all_data.groupby('Product')[numeric_cols].sum()

products = list(product_group.index)
result = product_group.sum(axis=1)

plt.bar(products, result)
plt.xticks(rotation='vertical', size=8)  # Corrected syntax
plt.xlabel('Product')
plt.ylabel('Quantity Ordered')
plt.title('Sum of Quantity Ordered by Product')
plt.show()  # Show the plot
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js