In [54]:
```
import pandas as pd
import os
```

Task 1: Merging 12 months of sales data into a single file

In [55]:
```
df = pd.read_csv("Sales Data/Sales_April_2019.csv")

files = [file for file in os.listdir("Sales Data")]

all_months_data = pd.DataFrame()

for file in files:
    df = pd.read_csv("Sales Data/"+file)
    all_months_data = pd.concat([all_months_data, df])

all_months_data.to_csv("all_data.csv", index = False)
```

Read in updated dataframe

In [56]:
```
all_data = pd.read_csv("all_data.csv")
all_data.head()
```

Out[56]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301

Task 2: Clean up the Data!

Drop rows of NAN

In [57]:
```
nan_df = all_data[all_data.isna().any(axis=1)]
nan_df.head()

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

Out[57]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301

Find 'Or' and delete it

In [58]:
```
all_data = all_data[all_data['Order Date'].str[0:2] != 'Or']
all_data.head()
```

Out[58]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301

Augment data with addition columns

In [59]:
```
all_data['Month'] = all_data['Order Date'].str[0:2]
all_data['Month'] = all_data['Month'].astype('int32')
all_data.head()
```

Out[59]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001	12
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001	12
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12

Add Sales Column

In [60]:
```
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

In [61]:
```
all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
all_data.sample(10)
```

Out[61]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
92745	203558	34in Ultrawide Monitor	1	379.99	05/01/19 18:49	336 Adams St, San Francisco, CA 94016	5	379.99
113797	292548	AAA Batteries (4-pack)	1	2.99	11/03/19 22:55	141 Church St, New York City, NY 10001	11	2.99
8300	303577	AA Batteries (4-pack)	1	3.84	12/13/19 17:00	838 5th St, Boston, MA 02215	12	3.84
59245	165537	Wired Headphones	4	11.99	03/13/19 18:02	472 Cherry St, Seattle, WA 98101	3	47.96
22563	317218	USB-C Charging Cable	1	11.95	12/06/19 06:42	684 Meadow St, Portland, ME 04101	12	11.95
64876	170930	Apple Airpods Headphones	1	150.00	03/01/19 13:21	554 Forest St, San Francisco, CA 94016	3	150.00
57191	163582	ThinkPad Laptop	1	999.99	03/03/19 13:12	303 Lake St, San Francisco, CA 94016	3	999.99
26488	177861	AA Batteries (4-pack)	1	3.84	04/23/19 06:32	712 Washington St, Boston, MA 02215	4	3.84
141349	144931	27in FHD Monitor	1	149.99	01/24/19 19:51	568 12th St, Los Angeles, CA 90001	1	149.99
89401	200388	Bose SoundSport Headphones	1	99.99	05/30/19 20:41	619 West St, New York City, NY 10001	5	99.99

Add a city column

In [62]:

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

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

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

Out[62]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New York City, NY
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New York City, NY
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95	New York City, NY
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco, CA
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta, GA

In [ ]:

Question 1: What was the best month for sales?

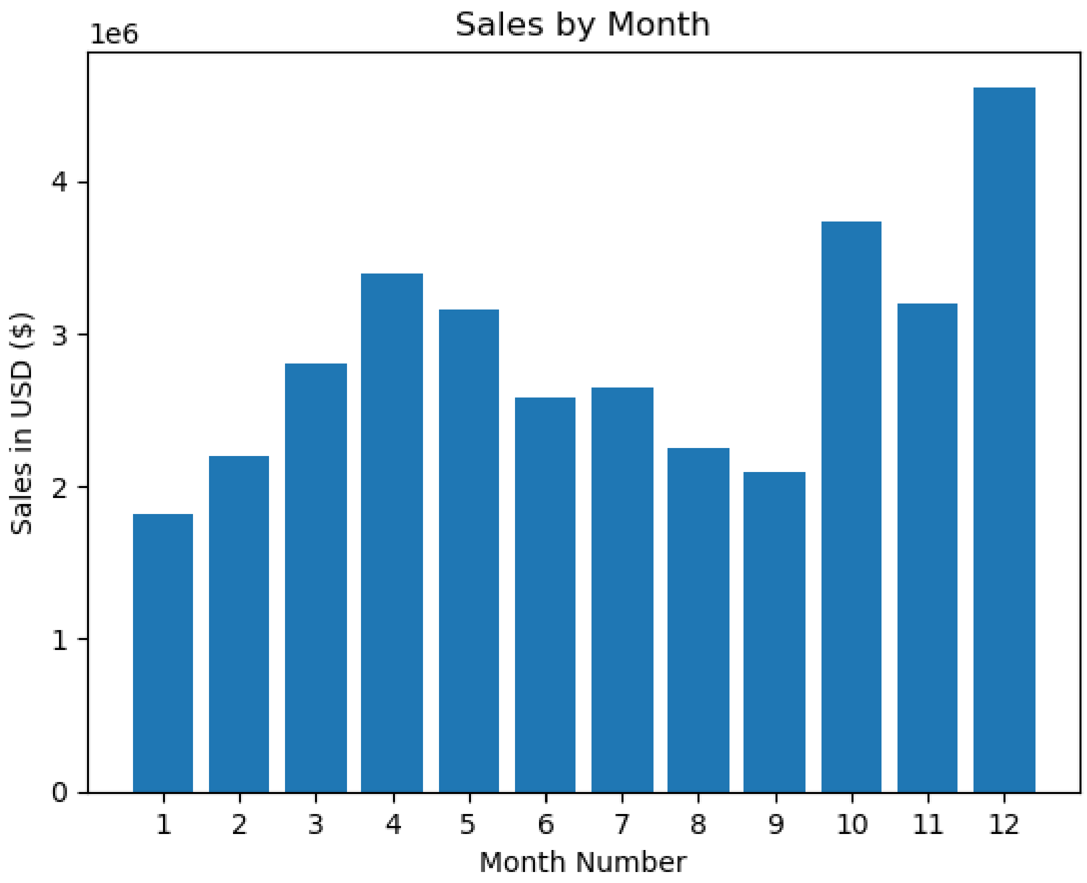
In [63]:
```
results = all_data.groupby('Month').sum()
```

In [64]:
```
import matplotlib.pyplot as plt
```

In [65]:
```
months = range(1,13)

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

Out[65]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



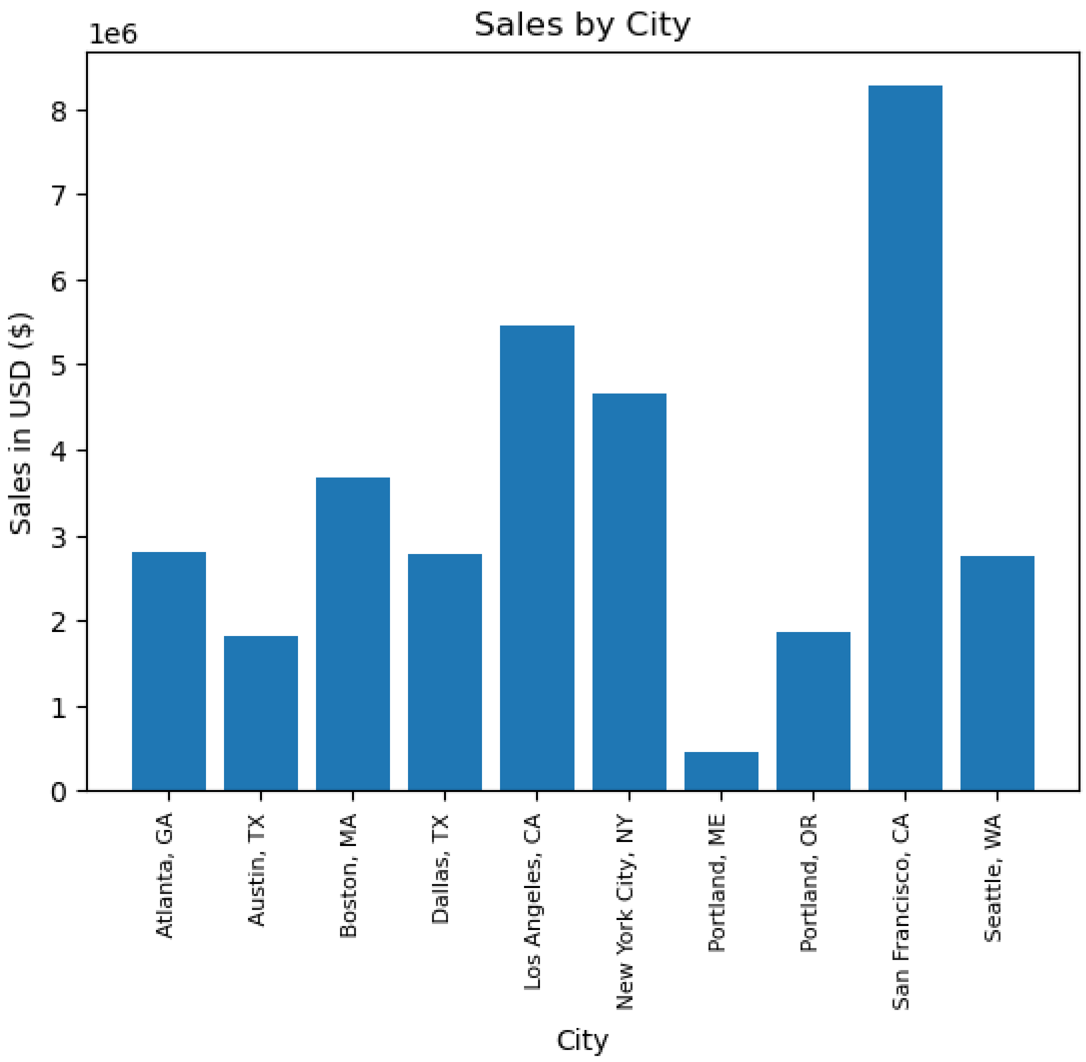
Question 2: What U.S. City had the highest number of sales

In [66]:

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

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')
plt.title('Sales by City')
plt.show()
```



Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?

In [67]:
```
all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

```
/var/folders/3w/rdvgs5053xz_4sgwf7mp_p300000gn/T/ipykernel_32187/3842191188.p
y:1: UserWarning: Could not infer format, so each element will be parsed indiv
idually, falling back to `dateutil`. To ensure parsing is consistent and as-ex
pected, please specify a format.
  all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

In [68]:
```
all_data['Hour'] = all_data['Order Date'].dt.hour
all_data['Minute'] = all_data['Order Date'].dt.minute
all_data.head()
```

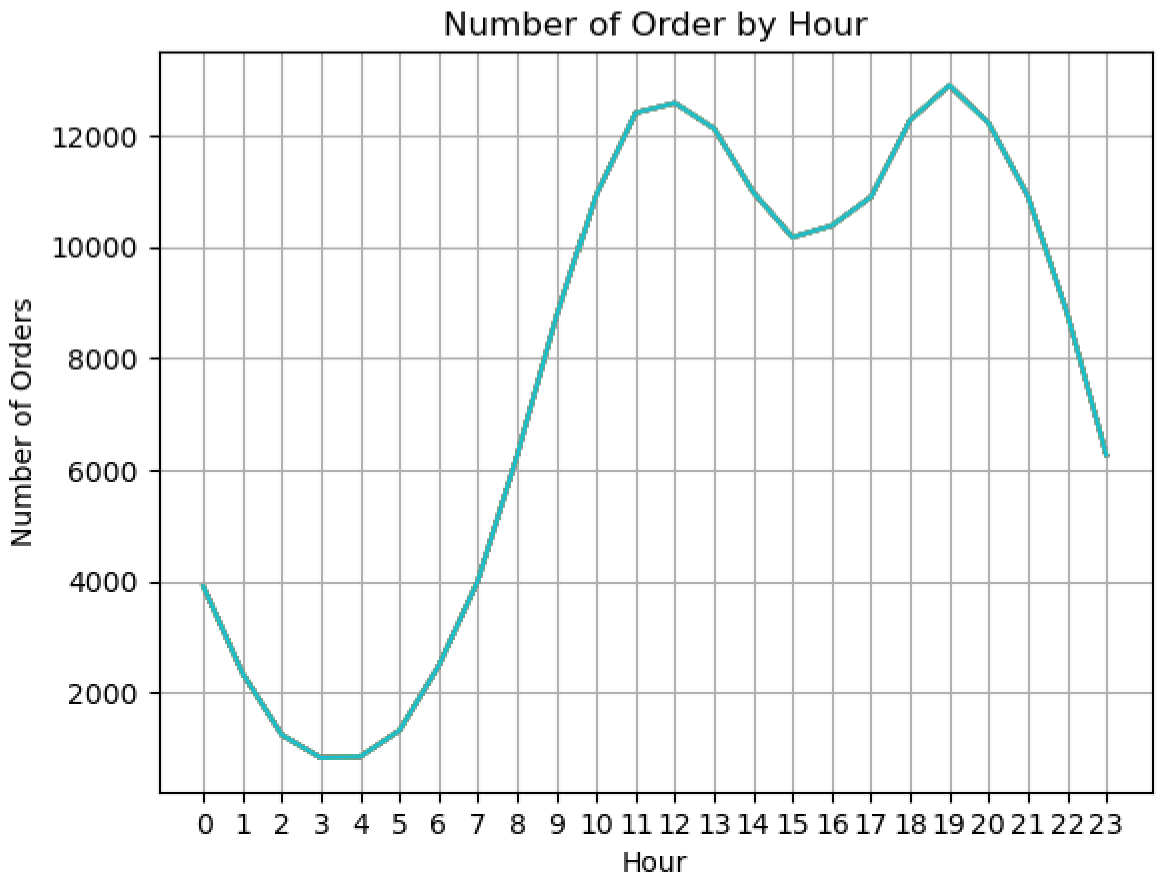
Out[68]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour
0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New York City, NY	0
1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New York City, NY	7
2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New York City, NY	18
3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco, CA	15
4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta, GA	12

In [69]:

```python
hours = [hour for hour, df in all_data.groupby('Hour')]

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



answer: Sales peek at 11am and 7pm, meaning we should display ad between 10am - 11pm and 6pm - 7pm.

Question 4: What products are most often sold together?

In [70]:
```
df = all_data[all_data['Order ID'].duplicated(keep=False)]
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join
df = df[['Order ID', 'Grouped']].drop_duplicates()
df.head()
```

```
/var/folders/3w/rdvgs5053xz_4sgwf7mp_p300000gn/T/ipykernel_32187/868064316.py:
2: 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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.jo
in(x))
```

Out[70]:

	Order ID	Grouped
16	295681	Google Phone,USB-C Charging Cable,Bose SoundSp...
36	295698	Vareebadd Phone,USB-C Charging Cable
42	295703	AA Batteries (4-pack),Bose SoundSport Headphones
66	295726	iPhone,Lightning Charging Cable
76	295735	iPhone,Apple Airpods Headphones,Wired Headphones

In [71]:

```python
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
```

answer: Top 3 products sold together: 1. iPhone and Lightning Cable 2. Google Phone and USB-C Cable 3. iPhone and Wired Headphones

Question 5: What product sold the most?

In [72]:

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

Out[72]:

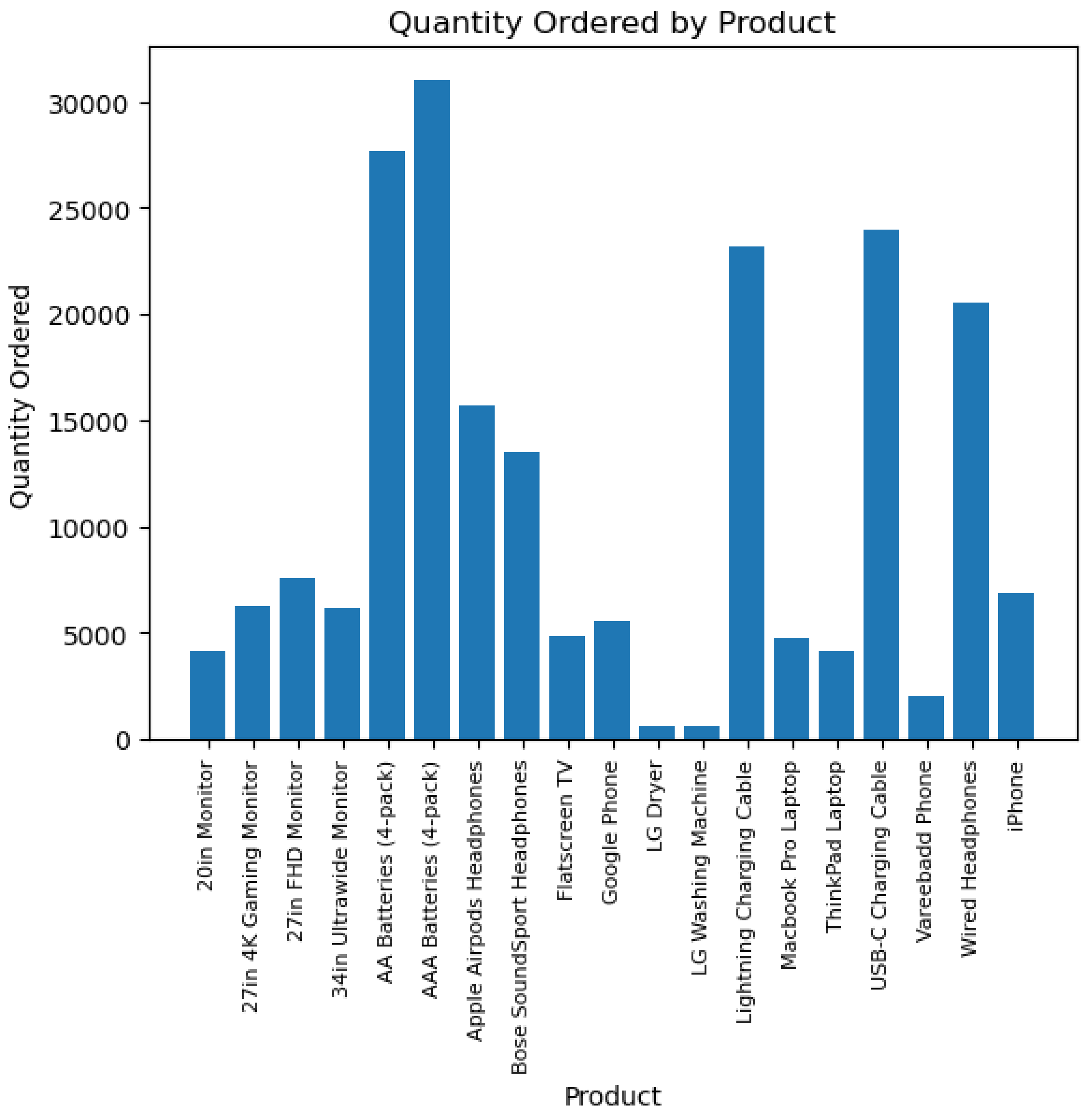
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour
0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New York City, NY	0
1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New York City, NY	7
2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New York City, NY	18
3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco, CA	15
4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta, GA	12

In [73]:

```python
#Groups the products and sums the quantity ordered
product_group = all_data.groupby('Product')
quantity_ordered = product_group[['Quantity Ordered']].sum()

products = [product for product, df in product_group]
quantities = quantity_ordered['Quantity Ordered']

# Creates a Bar Chart
plt.bar(products, quantities)
plt.xticks(rotation='vertical', size=8)
plt.xlabel('Product')
plt.ylabel('Quantity Ordered')
plt.title('Quantity Ordered by Product')
plt.show()
```



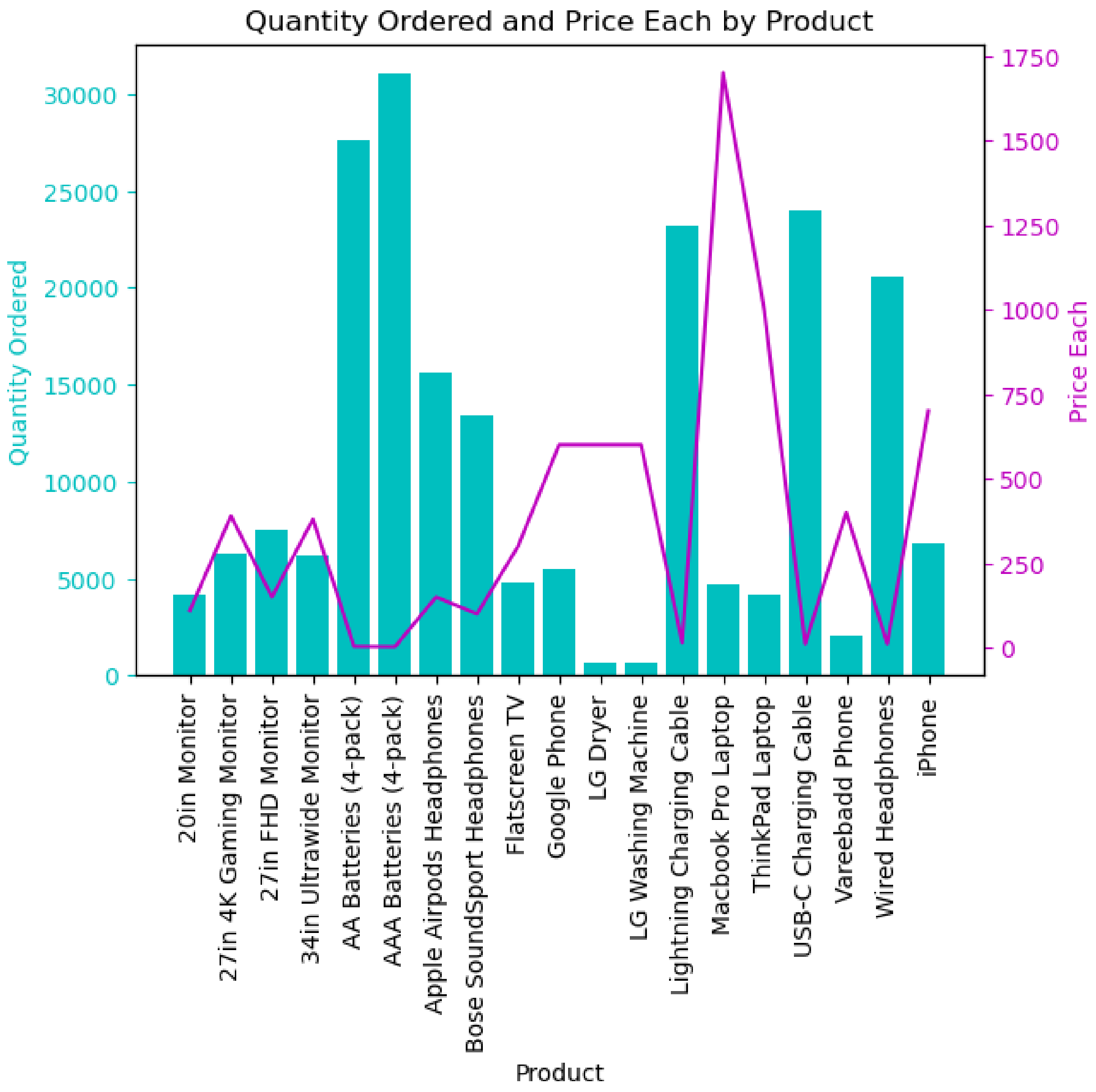
In [75]:
```
prices = product_group[['Price Each']].mean()

# Creates X and y axis
fig, ax1 = plt.subplots()
ax1.bar(products, quantities, color='c', label='Quantity Ordered')
ax1.set_xlabel('Product')
ax1.set_ylabel('Quantity Ordered', color='c')
ax1.tick_params('y', colors='c')

# Creates a second y-axis for prices
ax2 = ax1.twinx()
ax2.plot(products, prices, 'm-', label='Price Each')
ax2.set_ylabel('Price Each', color='m')
ax2.tick_params('y', colors='m')

# Rotating the x-axis labels vertically
ax1.tick_params(axis='x', rotation=90)

# Adds title
plt.title('Quantity Ordered and Price Each by Product')
plt.show()
```


Quantity Ordered and Price Each by Product
Quantity Ordered
0
5000
10000
15000
20000
25000
30000
Price Each
0
250
500
750
1000
1250
1500
1750
20in Monitor
27in 4K Gaming Monitor
27in FHD Monitor
34in Ultrawide Monitor
AA Batteries (4-pack)
AAA Batteries (4-pack)
Apple Airpods Headphones
Bose SoundSport Headphones
Flatscreen TV
Google Phone
LG Dryer
LG Washing Machine
Lightning Charging Cable
Macbook Pro Laptop
ThinkPad Laptop
USB-C Charging Cable
Vareebadd Phone
Wired Headphones
iPhone
Product

```
In [ ]:
```