```
import requests
from bs4 import BeautifulSoup

# Send a GET request to the URL
url = https://www.amazon.in/s?k-bags&crid=2M096C6104MLT&gid=1653308124&sprefix ba,aps%2C283&ref=sr_pg 1
response = requests.get(url)

# Parse the HTML content of the page using BeautifulSoup
soup = BeautifulSoup(response.content, 'html.parser')

# Loop through the products and extract the required information
for product in soup.find_all('div', {'class': 's-result-item'}):
    # Extract product URL
    product_url = product.find('a', {'class': 'a-link-normal'})['href']

    # Extract product name
    product_name = product.find('h2', {'class': 'a-size-mini a-spacing-none a-color-base s-line-clamp-2'}).text.strip()

    # Extract product price
    product_price = product.find('span', {'class': 'a-price-whole'}).text.strip()

    # Extract product rating
    product_rating = product.find('span', {'class': 'a-icon-alt'}).text.strip()

    # Extract number of reviews
    product_reviews = product.find('span', {'class': 'a-size-base'}).text.strip()
```

```
    # Print the extracted information
    print(f"Product URL: {product_url}")
    print(f"Product Name: {product_name}")
    print(f"Product Price: {product_price}")
    print(f"Product Rating: {product_rating}")
    print(f"Number of Reviews: {product_reviews}")
    print()
```

Solution 2

```
import csv
import requests
from bs4 import BeautifulSoup
import time

def scrape_product_data(product_url):
    “””
    Scrapes the product data from a given product URL and returns it as a dictionary.
    “””
    Response = requests.get(product_url)
    Soup = BeautifulSoup(response.content, ‘html.parser’)

    Product_title = soup.find(‘span’, {‘id’: ‘productTitle’}).text.strip()

    Try:
        Product_price = soup.find(‘span’, {‘class’: ‘a-price-whole’}).text.strip()
    Except AttributeError:
        Product_price = ‘N/A’

    Try:
        Product_rating = soup.find(‘span’, {‘class’: ‘a-icon-alt’}).text.strip().split()[0]
```

```
Except AttributeError:
    Product_rating = 'N/A'

Try:
    Num_reviews = soup.find('span', {'id': 'acrCustomerReviewText'}).text.strip().split()[0]
Except AttributeError:
    Num_reviews = 'N/A'

Try:
    Asin = soup.find('div', {'data-feature-name': 'product-asin'}).get('data-asin')
Except AttributeError:
    Asin = 'N/A'

Try:
    Product_description = soup.find('div', {'id': 'productDescription'}).text.strip()
Except AttributeError:
    Product_description = 'N/A'

Try:
    Manufacturer = soup.find('a', {'id': 'bylineInfo'}).text.strip()
Except AttributeError:
    Manufacturer = 'N/A'

Product_data = {
    'Product URL': product_url,
    'Product Name': product_title,
    'Product Price': product_price,
    'Rating': product_rating,
    'Number of reviews': num_reviews,
```

```
        'ASIN': asin,
        'Product Description': product_description,
        'Manufacturer': manufacturer
    }

    Return product_data

# Scrape product data from multiple pages
Base_url = 'https://www.amazon.in/sk-rid-2M096C104MIT/ref=sr_pg_{}?qid=16533083246rf-Ca&pageNumber={}'
Products = []
For page in range(1, 21):
    url = base_url.format(page, page)
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')
    product_links = soup.find_all('a', {'class': 'a-link-normal s-no-outline'})

    for link in product_links:
        product_url = 'https://www.amazon.in' + link.get('href')
        product_data = scrape_product_data(product_url)
        products.append(product_data)
        time.sleep(1) # Add delay to avoid being blocked by Amazon

# Export product data to CSV file
With open('amazon_products.csv', 'w', newline='', encoding='utf-8') as file:
    Writer = csv.DictWriter(file, fieldnames=product_data.keys())
    Writer.writeheader()
    For product in products:
        Writer.writerow(product)
```