Ans 1)

```
import requests
from bs4 import BeautifulSoup

def search_amazon(product):
    base_url = 'https://www.amazon.in'
    search_url = f'{base_url}/s?k={product.replace(" ", "+")}'

    response = requests.get(search_url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.text, 'html.parser')
        products = soup.find_all('div', {'data-component-type': 's-search-result'})

        if products:
            for product in products:
                product_title = product.find('span', {'class': 'a-text-normal'}).text.strip()
                product_price = product.find('span', {'class': 'a-price-whole'}).text.strip()
                print(f'{product_title} - ₹{product_price}')
        else:
            print('No products found.')
    else:
        print('Failed to connect to Amazon.in')

if __name__ == "__main__":
    user_input = input("Enter the product you want to search for on Amazon.in: ")
    search_amazon(user_input)
```

Ans 2)

```
import requests
import pandas as pd
from bs4 import BeautifulSoup

def scrape_product_details(product):
    base_url = 'https://www.amazon.in'
    search_url = f'{base_url}/s?k={product.replace(" ", "+")}'
    all_products = []

    for page_num in range(1, 4):  # Scrape first 3 pages
        page_url = f'{search_url}&page={page_num}'
```

```
response = requests.get(page_url)
if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')
    products = soup.find_all('div', {'data-component-type': 's-search-result'})

    if products:
        for product in products:
            details = {}
            product_title = product.find('span', {'class': 'a-text-normal'})
            if product_title:
                details['Name of the Product'] = product_title.text.strip()
            else:
                details['Name of the Product'] = '-'

            product_brand = product.find('span', {'class': 'a-size-base-plus'})
            if product_brand:
                details['Brand Name'] = product_brand.text.strip()
            else:
                details['Brand Name'] = '-'

            product_price = product.find('span', {'class': 'a-price-whole'})
            if product_price:
                details['Price'] = product_price.text.strip()
            else:
                details['Price'] = '-'

            product_return = product.find('span', {'class': 'a-icon-alt'})
            if product_return:
                details['Return/Exchange'] = product_return.text.strip()
            else:
                details['Return/Exchange'] = '-'

            product_delivery = product.find('span', {'class': 'a-text-bold'})
            if product_delivery:
                details['Expected Delivery'] = product_delivery.text.strip()
            else:
                details['Expected Delivery'] = '-'

            product_availability = product.find('span', {'class': 'a-size-base'})
            if product_availability:
                details['Availability'] = product_availability.text.strip()
```

```
                else:
                    details['Availability'] = '-'

                product_link = product.find('a', {'class': 'a-link-normal'})
                if product_link:
                    details['Product URL'] = base_url + product_link['href']
                else:
                    details['Product URL'] = '-'

                all_products.append(details)
            else:
                print('No products found on page', page_num)
        else:
            print('Failed to connect to Amazon.in')

    return all_products

def save_to_csv(product_details, filename):
    df = pd.DataFrame(product_details)
    df.to_csv(filename, index=False)
    print(f'Data saved to {filename} successfully.')

if __name__ == "__main__":
    user_input = input("Enter the product you want to search for on Amazon.in: ")
    product_details = scrape_product_details(user_input)
    save_to_csv(product_details, 'amazon_products.csv')
```

Ans 3)

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import os
import requests
from bs4 import BeautifulSoup

def scrape_images(keyword, num_images):
    driver = webdriver.Chrome()  # Provide path to your chromedriver
    driver.get("https://www.google.com/imghp?hl=en")
```

```
    search_box = driver.find_element_by_xpath("//input[@name='q']")
    search_box.send_keys(keyword)
    search_box.send_keys(Keys.RETURN)

    # Scroll to load more images
    last_height = driver.execute_script("return document.body.scrollHeight")
    while True:
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
        time.sleep(2)
        new_height = driver.execute_script("return document.body.scrollHeight")
        if new_height == last_height:
            break
        last_height = new_height

    # Parse the HTML
    soup = BeautifulSoup(driver.page_source, 'html.parser')
    image_tags = soup.find_all('img', {'class': 'rg_i'})

    # Create a directory to save images
    if not os.path.exists(keyword):
        os.makedirs(keyword)

    # Download images
    for i, image_tag in enumerate(image_tags[:num_images]):
        image_url = image_tag.get('src')
        if image_url:
            image_path = os.path.join(keyword, f"{keyword}_{i+1}.jpg")
            with open(image_path, 'wb') as f:
                f.write(requests.get(image_url).content)
            print(f"Downloaded image {i+1} for '{keyword}'")
        if i+1 == num_images:
            break

    driver.quit()

if __name__ == "__main__":
    keywords = ['fruits', 'cars', 'Machine Learning', 'Guitar', 'Cakes']
    num_images_per_keyword = 10
    for keyword in keywords:
        scrape_images(keyword, num_images_per_keyword)
```

Ans 4)

```
import pandas as pd
import requests
from bs4 import BeautifulSoup

def scrape_smartphone_details(product):
    base_url = 'https://www.flipkart.com'
    search_url = f'{base_url}/search?q={product.replace(" ", "+")}'

    response = requests.get(search_url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.text, 'html.parser')
        products = soup.find_all('div', {'class': '_1AtVbE'})

        all_smartphones = []

        for product in products:
            details = {}

            product_title = product.find('div', {'class': '_4rR01T'})
            if product_title:
                details['Smartphone name'] = product_title.text.strip()
            else:
                details['Smartphone name'] = '-'

            product_brand = product_title.text.split()[0]
            details['Brand Name'] = product_brand if product_brand else '-'

            product_color = product.find('div', {'class': '_2WkVRV'})
            details['Colour'] = product_color.text.strip() if product_color else '-'

            specifications = product.find_all('li', {'class': 'rgWa7D'})
            for spec in specifications:
                if 'RAM' in spec.text:
                    details['RAM'] = spec.text.split()[0] if spec.text.split()[0] else '-'
                elif 'ROM' in spec.text:
                    details['Storage(ROM)'] = spec.text.split()[0] if spec.text.split()[0] else '-'
                elif 'Primary Camera' in spec.text:
                    details['Primary Camera'] = spec.text.split()[0] if spec.text.split()[0] else '-'
```

```
            elif 'Secondary Camera' in spec.text:
                details['Secondary Camera'] = spec.text.split()[0] if spec.text.split()[0] else '-'
            elif 'Display Size' in spec.text:
                details['Display Size'] = spec.text.split()[0] if spec.text.split()[0] else '-'
            elif 'Battery Capacity' in spec.text:
                details['Battery Capacity'] = spec.text.split()[0] if spec.text.split()[0] else '-'

        product_price = product.find('div', {'class': '_30jeq3 _1_WHN1'})
        details['Price'] = product_price.text.strip() if product_price else '-'

        product_link = product.find('a', {'class': '_1fQZEK'})
        details['Product URL'] = base_url + product_link['href'] if product_link else '-'

        all_smartphones.append(details)

    return all_smartphones
  else:
    print('Failed to connect to Flipkart.com')
    return []

def save_to_csv(data, filename):
  df = pd.DataFrame(data)
  df.to_csv(filename, index=False)
  print(f'Data saved to {filename} successfully.')

if __name__ == "__main__":
  user_input = input("Enter the smartphone you want to search for on Flipkart: ")
  smartphone_details = scrape_smartphone_details(user_input)
  save_to_csv(smartphone_details, 'flipkart_smartphones.csv')
```

Ans 5)

```
import requests

def get_coordinates(city):
  api_key = 'YOUR_GOOGLE_MAPS_API_KEY'  # Replace with your Google Maps API key
  url = f'https://maps.googleapis.com/maps/api/geocode/json?address={city}&key={api_key}'

  response = requests.get(url)
  if response.status_code == 200:
```

```
        data = response.json()
        if data['status'] == 'OK':
            location = data['results'][0]['geometry']['location']
            latitude = location['lat']
            longitude = location['lng']
            return latitude, longitude
        else:
            print(f"Failed to find coordinates for {city}")
    else:
        print("Failed to connect to Google Maps API")

if __name__ == "__main__":
    city = input("Enter the city name: ")
    coordinates = get_coordinates(city)
    if coordinates:
        print(f"Coordinates for {city}: Latitude = {coordinates[0]}, Longitude = {coordinates[1]}")
```

Ans 6)

```
import requests
from bs4 import BeautifulSoup

def scrape_gaming_laptops():
    url = 'https://www.digit.in/top-products/best-gaming-laptops-40.html'
    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.text, 'html.parser')
        laptops = soup.find_all('div', class_='TopNumbeHeading active')
        details = []

        for laptop in laptops:
            laptop_details = {}
            laptop_details['Name'] = laptop.find('div', class_='TopNumbeHeading active').text.strip()
            laptop_details['Price'] = laptop.find('td', class_='smprice').text.strip()
            laptop_details['Specification'] = laptop.find('td', class_='smtd').text.strip()
            details.append(laptop_details)

        return details
    else:
```

```
        print('Failed to fetch data from digit.in')

if __name__ == "__main__":
    gaming_laptops = scrape_gaming_laptops()
    for laptop in gaming_laptops:
        print(laptop)
```

Ans 7)

```
import requests
from bs4 import BeautifulSoup

def scrape_billionaires():
    url = 'https://www.forbes.com/billionaires/'
    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        billionaires = soup.find_all('div', class_='personName')

        details = []
        for billionaire in billionaires:
            info = {}
            info['Rank'] = billionaire.find_previous('div', class_='rank').text.strip()
            info['Name'] = billionaire.text.strip()
            details.append(info)

        net_worths = [item.text.strip() for item in soup.find_all('div', class_='netWorth')]
        ages = [item.text.strip() for item in soup.find_all('div', class_='age')]
        citizenships = [item.text.strip() for item in soup.find_all('div',
class_='countryOfCitizenship')]
        sources = [item.text.strip() for item in soup.find_all('div', class_='source-column')]
        industries = [item.text.strip() for item in soup.find_all('div', class_='category')]

        for i, detail in enumerate(details):
            detail['Net worth'] = net_worths[i]
            detail['Age'] = ages[i]
            detail['Citizenship'] = citizenships[i]
            detail['Source'] = sources[i]
            detail['Industry'] = industries[i]
```

```
        return details
    else:
        print('Failed to fetch data from www.forbes.com')

if __name__ == "__main__":
    billionaires_data = scrape_billionaires()
    for billionaire in billionaires_data:
        print(billionaire)
```

Ans 8)

```
import os
import googleapiclient.discovery

# Set up API key
api_key = 'YOUR_YOUTUBE_API_KEY'

def get_video_id(video_url):
    # Extract video ID from URL
    video_id = video_url.split('=')[-1]
    return video_id

def get_video_comments(video_id):
    youtube = googleapiclient.discovery.build('youtube', 'v3', developerKey=api_key)

    # Retrieve comments for the video
    comments = []
    next_page_token = None
    while len(comments) < 500:
        request = youtube.commentThreads().list(
            part='snippet',
            videoId=video_id,
            maxResults=min(100, 500 - len(comments)),
            pageToken=next_page_token
        )
        response = request.execute()

        for item in response['items']:
```

```
            comment = {}
            comment['comment'] = item['snippet']['topLevelComment']['snippet']['textOriginal']
            comment['upvotes'] = item['snippet']['topLevelComment']['snippet']['likeCount']
            comment['time'] = item['snippet']['topLevelComment']['snippet']['publishedAt']
            comments.append(comment)

        next_page_token = response.get('nextPageToken')

        if not next_page_token:
            break

    return comments

if __name__ == "__main__":
    video_url = input("Enter the URL of the YouTube video: ")
    video_id = get_video_id(video_url)
    comments = get_video_comments(video_id)

    print(f"Total comments extracted: {len(comments)}")
    for comment in comments:
        print(f"Comment: {comment['comment']}")
        print(f"Upvotes: {comment['upvotes']}")
        print(f"Time: {comment['time']}")
        print("-" * 50)
```

We have to replace 'YOUR_YOUTUBE_API_KEY' with our actual YouTube Data API v3 key.

Ans 9)

```
import requests
from bs4 import BeautifulSoup

def scrape_hostels(location):
    url = f'https://www.hostelworld.com/findabed.php/ChosenCity.{location}/
ChosenCountry.England'
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'}

    response = requests.get(url, headers=headers)
```

```
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        hostels = soup.find_all('div', class_='fabresult rounded clearfix')

        hostel_data = []
        for hostel in hostels:
            data = {}
            data['Name'] = hostel.find('h2', class_='title-6').text.strip()
            data['Distance from City Centre'] = hostel.find('span', class_='description').text.strip()
            data['Ratings'] = hostel.find('div', class_='score orange big').text.strip()
            data['Total Reviews'] = hostel.find('div', class_='reviews').text.strip().split()[0]
            data['Overall Reviews'] = hostel.find('div', class_='keyword').text.strip()
            data['Privates from Price'] = hostel.find('span', class_='price title-5').text.strip()
            data['Dorms from Price'] = hostel.find('span', class_='price title-5').text.strip()
            data['Facilities'] = ', '.join([facility.text.strip() for facility in hostel.find_all('span',
class_='facilities-label')])
            data['Description'] = hostel.find('div', class_='rating-factors prop-info').text.strip()
            hostel_data.append(data)

        return hostel_data
    else:
        print('Failed to fetch data from www.hostelworld.com')

if __name__ == "__main__":
    location = 'London'
    hostels_data = scrape_hostels(location)
    for hostel in hostels_data:
        print(hostel)
```