```
import requests

from bs4 import BeautifulSoup

from urllib.parse import quote, unquote

import re

import pandas as pd  # For saving to Excel and CSV




# Base URL

base_url = "https://www.ninisite.com/clinic/tag/questions/"




# Full list of hashtags and their IDs

hashtags = [

    {"id": 4951, "hashtag": "مشاوره-خانواد", "samples": 50},

    {"id": 6522, "hashtag": "روانشناسی", "samples": 50},
```

{"id": 10231, "hashtag": "ازدواج", "samples": 50},

{"id": 4694, "hashtag": "کودک-تربیت", "samples": 50},

{"id": 7674, "hashtag": "دوستی", "samples": 50},

{"id": 2848, "hashtag": "ارتباط-اجتماعی", "samples": 50},

{"id": 232, "hashtag": "اعتماد-به-نفس", "samples": 50},

{"id": 1784, "hashtag": "خجالت", "samples": 50},

{"id": 1792, "hashtag": "اضطراب", "samples": 50},

{"id": 1015, "hashtag": "استرس", "samples": 50},

{"id": 407, "hashtag": "اختلال", "samples": 50},

{"id": 3659, "hashtag": "اختلال-دوقطبی", "samples": 50},

{"id": 89, "hashtag": "افسردگی", "samples": 50},

{"id": 2189, "hashtag": "پوچی", "samples": 50},

{"id": 352, "hashtag": "زندگی-مشترک", "samples": 50}, {"id": 1647, "hashtag": "بی-انگیزه", "samples": 50},

{"id": 555, "hashtag": "ناامید", "samples": 50},

{"id": 761, "hashtag": "خودکشی", "samples": 50},

{"id": 289, "hashtag": "طلاق", "samples": 50},

{"id": 362, "hashtag": "اعتیاد", "samples": 50},

{"id": 8966, "hashtag": "ترک", "samples": 50},

{"id": 42, "hashtag": "خودارضایی", "samples": 50},

{"id": 2074, "hashtag": "انزال", "samples": 50},

{"id": 28, "hashtag": "خانواده", "samples": 50},

{"id": 3478, "hashtag": "حس-تنهایی", "samples": 50},

{"id": 6851, "hashtag": "نوجوانی", "samples": 50},

{"id": 917, "hashtag": "حساسیت", "samples": 50},

{"id": 403, "hashtag": "خیانت", "samples": 50}, {"id": 1791, "hashtag": "پانیک", "samples": 50},

{"id": 1622, "hashtag": "وسواس-فکری", "samples": 50},

{"id": 1277, "hashtag": "استرس-وترس", "samples": 50},

{"id": 2, "hashtag": "کودک", "samples": 50},

{"id": 3496, "hashtag": "کنکور", "samples": 50},

{"id": 590, "hashtag": "مادرشوهر", "samples": 50},

{"id": 827, "hashtag": "طلاق-عاطفی", "samples": 50},

{"id": 2317, "hashtag": "دوران-عقد", "samples": 50},

{"id": 2037, "hashtag": "مردخسیس", "samples": 50},

{"id": 9271, "hashtag": "خرجی-ندادن-شوهر", "samples": 50},

{"id": 920, "hashtag": "همسرداری", "samples": 50},

{"id": 5214, "hashtag": "تهدید", "samples": 50},

{"id": 2773, "hashtag": "خواهر", "samples": 50},

{"id": 8952, "hashtag": "بدرفتاری", "samples": 50},

{"id": 257, "hashtag": "عصبی", "samples": 50},

{"id": 540, "hashtag": "رفتار-همسر", "samples": 50},

{"id": 17, "hashtag": "کودک-من", "samples": 50},

{"id": 607, "hashtag": "خانواده-همسر", "samples": 50},

{"id": 160, "hashtag": "مشکلات-بعد-ازدواج", "samples": 50},

{"id": 1432, "hashtag": "خانواده-مادری", "samples": 50},

{"id": 1404, "hashtag": "وسواس", "samples": 50},

{"id": 12337, "hashtag": "روانپزشک", "samples": 50},

{"id": 8992, "hashtag": "عادت-بد", "samples": 50},

{"id": 118, "hashtag": "کودکان", "samples": 50},

{"id": 4741, "hashtag": "غر-زدن", "samples": 50},

{"id": 10566, "hashtag": "بهانه-گیری", "samples": 50},

{"id": 153, "hashtag": "خشم", "samples": 50},

{"id": 2710, "hashtag": "پشیمانی", "samples": 50},

{"id": 330, "hashtag": "خواستگار", "samples": 50},

{"id": 453, "hashtag": "عقد", "samples": 50},

{"id": 3872, "hashtag": "دوست-داشتن", "samples": 50},

{"id": 4276, "hashtag": "عدم-تمرکز", "samples": 50},

{"id": 11804, "hashtag": "اضطراب-اجتماعی", "samples": 50},

{"id": 10696, "hashtag": "شکست-عشقی", "samples": 50},

{"id": 3079, "hashtag": "پارانویید", "samples": 50},

{"id": 2267, "hashtag": "پدر", "samples": 50},

{"id": 2361, "hashtag": "فوت", "samples": 50},

{"id": 2711, "hashtag": "ازدواج-مجدد", "samples": 50},

{"id": 5163, "hashtag": "تصمیم-گیری", "samples": 50}, {"id": 1098, "hashtag": "وابستگی", "samples": 50},

{"id": 4898, "hashtag": "آسیب", "samples": 50},

{"id": 65, "hashtag": "بکارت", "samples": 50},

{"id": 279, "hashtag": "مادر", "samples": 50},

{"id": 2484, "hashtag": "پرخاشگری", "samples": 50},

{"id": 1655, "hashtag": "فحش", "samples": 50},

{"id": 1063, "hashtag": "کتک-زدن", "samples": 50},

{"id": 1753, "hashtag": "خیانت-همسر", "samples": 50},

{"id": 113, "hashtag": "همسر", "samples": 50},

{"id": 4673, "hashtag": "بدبین", "samples": 50},

{"id": 7304, "hashtag": "مواد-مخدر", "samples": 50},

{"id": 321, "hashtag": "شک", "samples": 50},

{"id": 546, "hashtag": "مشکلات-رابطه-ای-باهمسری", "samples": 50},

```
    {"id": 1321, "hashtag": "اینترنت", "samples": 50},

    {"id": 1053, "hashtag": "تربیت-جنسی", "samples": 50},

    {"id": 333, "hashtag": "ترس", "samples": 50},

]

# Function to check if a page exists

def page_exists(url):

    try:

        response = requests.get(url)

        if response.status_code == 200:

            return response.text  # Return the page content if it exists

        return None

    except requests.exceptions.RequestException as e:

        print(f"Error checking URL {url}: {e}")
        return None



# Function to scrape question links from a page

def scrape_question_links(page_content):

    soup = BeautifulSoup(page_content, "html.parser")

    question_links = []



    # Find all links that match the question format

    for link in soup.find_all("a", href=True):

        href = link["href"]

        # Match the format: /clinic/question/<id>/<slug>

        if re.match(r"^/clinic/question/\d+/.+", href):

            full_url = f"https://www.ninisite.com{href}"  # Construct the full URL
            question_links.append(full_url)  # Append the full URL with Farsi slug

    return question_links

# Main script to gather links grouped by topic
all_data = []  # List to store all data for Excel/CSV
for tag in hashtags:
    encoded_hashtag = quote(tag["hashtag"])  # Encode the hashtag
    page = 1  # Start with page 1
    topic_links = []  # List to store links for this topic
    max_pages = 50  # Set a maximum number of pages to scrape
    print(f"\nProcessing topic: {tag['hashtag']} (ID: {tag['id']})")
    while len(topic_links) < tag["samples"] and page <= max_pages:  # Stop once we have the required number of links or reach max pages

        url = f"{base_url}{tag['id']}/{encoded_hashtag}?page={page}"  # Construct the URL

        page_content = page_exists(url)  # Check if the page exists and get its content


        if page_content:

            links = scrape_question_links(page_content)  # Scrape question links from the page

            if links:

                topic_links.extend(links)  # Add new links to the topic list

                print(f"Collected {len(links)} links from {url}")

            else:

                print(f"No question links found on {url}")

                break  # Stop if no links are found on the page
            page += 1  # Move to the next page

        else:

            print(f"End of pages for topic: {tag['hashtag']}")

            break  # Exit the loop when no more pages exist




    # Add topic and its links to the data

     all_data.append({"Topic": tag["hashtag"], "ID": tag["id"], "Links": None, "Direct URL": None})  # Add topic
and ID as a row

    for i, link in enumerate(topic_links[:tag["samples"]], start=1):  # Add links under the topic

        decoded_link = unquote(link)  # Decode the link to Farsi

        all_data.append({

            "Topic": None,

            "ID": None,

            "Links": f"{i}: {decoded_link}",
            "Direct URL": decoded_link  # Add the direct clickable URL

        })



# Convert the data to a DataFrame

df = pd.DataFrame(all_data)



# Save to Excel

excel_file = "question_links_grouped_with_urls.xlsx"

df.to_excel(excel_file, index=False)

print(f"\nSaved links to Excel file: {excel_file}")



# Save to CSV

csv_file = "question_links_grouped_with_urls.csv"
```

```
df.to_csv(csv_file, index=False)

print(f"Saved links to CSV file: {csv_file}")



# Output the results

print(f"\nCollected links grouped by topics.")

for row in all_data:

    if row["Topic"]:

        print(f"\nTopic: {row['Topic']} (ID: {row['ID']})")

    elif row["Links"]:

        print(row["Links"])
```