1. **Write a program Magic 8 ball with list.**

```
import random


def magic8ball():
   response = input("(Press 'any key' for answer and 'quit' to exit)\nWhat is your question?\n")
    Eightball_answers = [
        "It is certain",
        "Outlook good",
        "You may rely on it",
        "Ask again later",
        "Concentrate and ask again",
        "Reply hazy, try again",
        "My reply is no",
        "My sources say no"
        ]

    if response == "quit":
        print("Have A Good Day!")
    else:
        print(random.choice(Eightball_answers), "\n")
        magic8ball()

magic8ball()
```

2. Write a password locker program.

```
import random
import array

# maximum length of password needed
MAX_LEN = 12

# declare arrays of the character that we need in out password
DIGITS = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
LOCASE_CHARACTERS = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
                     'i', 'j', 'k', 'm', 'n', 'o', 'p', 'q',
                     'r', 's', 't', 'u', 'v', 'w', 'x', 'y','z']

UPCASE_CHARACTERS = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
                     'I', 'J', 'K', 'M', 'N', 'O', 'P', 'Q',
                     'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y','Z']

SYMBOLS = ['@', '#', '$', '%', '=', ':', '?', '.', '/', '|', '~', '>',
           '*', '(', ')', '<']

# combines all the character arrays above to form one array
COMBINED_LIST = DIGITS + UPCASE_CHARACTERS + LOCASE_CHARACTERS + SYMBOLS

# randomly select at least one character from each character set above
rand_digit = random.choice(DIGITS)
rand_upper = random.choice(UPCASE_CHARACTERS)
rand_lower = random.choice(LOCASE_CHARACTERS)
rand_symbol = random.choice(SYMBOLS)

# combine the character randomly selected above
temp_pass = rand_digit + rand_upper + rand_lower + rand_symbol

# now that we are sure we have at least one character from each
for x in range(MAX_LEN - 4):
        temp_pass = temp_pass + random.choice(COMBINED_LIST)

        # convert temporary password into array and shuffle to
        temp_pass_list = array.array('u', temp_pass)
        random.shuffle(temp_pass_list)

# traverse the temporary password array and append the chars
for x in temp_pass_list:
            password = password + x

# print out password
print(password)
```

3. **Write a program to extract phone number and email address using Regex.**

```
import re
def extract_phone_numbers_and_emails(text):

 # Define regex patterns for phone numbers and emails
  phone_pattern = re.compile(r'\b(\d{3}[-.\s]?\d{3}[-.\s]?\d{4})\b')
 email_pattern = re.compile(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b')

    # Find all matches in the text
    phone_numbers = re.findall(phone_pattern, text)
    emails = re.findall(email_pattern, text)
    return phone_numbers, emails
    phone_numbers, emails = extract_phone_numbers_and_emails(text_to_search)
    print("Phone Numbers:", phone_numbers)
    print("Emails:", emails)
```

4. **Program to generate random quiz file of guessing state capitals.**

```
import random

def generate_quiz(quiz_filename, states_and_capitals):

    # Shuffle the order of states
    states = list(states_and_capitals.keys())
    random.shuffle(states)

    # Create a quiz file
    with open(quiz_filename, 'w') as quiz_file:

        # Write the header
        quiz_file.write("Guess the State Capitals Quiz\n\n")

        # Generate questions and write to the file
        for i, state in enumerate(states, start=1):
            capital = states_and_capitals[state]
            options = list(states_and_capitals.values())
            options.remove(capital)
            options = random.sample(options, 3)
            options.append(capital)
            random.shuffle(options)

            # Write the question
            quiz_file.write(f"{i}. What is the capital of {state}?\n")

            # Write the options
            for j, option in enumerate(options, start=1):
                quiz_file.write(f"   {j}. {option}\n")
                quiz_file.write("\n")
```

```
    # Write the answer key
        quiz_file.write(f"Answer: {options.index(capital) + 1}\n\n")

# Example usage
    states_and_capitals = {
     "Alabama": "Montgomery",
     "Alaska": "Juneau",
     # Add more states and capitals as needed
}

quiz_filename = "state_capitals_quiz.txt"
generate_quiz(quiz_filename, states_and_capitals)
print(f"Quiz generated and saved to {quiz_filename}")
```

5. **Write a program to download all XKCD Comics to a drive.**

```
import os
import requests
from bs4 import BeautifulSoup

def download_xkcd_comics(output_folder):
    # Create output folder if it doesn't exist
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    # Start from the first comic
    current_comic_num = 1

    # Continue downloading until a comic is not found
    while True:

        # Construct the URL for the current comic
        url = f'https://xkcd.com/{current_comic_num}/'

        # Make a request to the XKCD website
        response = requests.get(url)

        # Check if the comic exists
        if response.status_code == 404:
            print(f"Comic {current_comic_num} not found. Stopping download.")
            break

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

        # Find the image URL in the HTML
        img_tag = soup.find('div', {'id': 'comic'}).find('img')
        if img_tag:
            img_url = f'https:{img_tag["src"]}'
            img_response = requests.get(img_url)
```

```
            # Save the image to the output folder
with open(os.path.join(output_folder, f'xkcd_{current_comic_num}.png'), 'wb') as img_file:
                    img_file.write(img_response.content)
                print(f"Downloaded comic {current_comic_num}")

                # Move to the next comic
                current_comic_num += 1

        # Example usage
        output_folder = 'xkcd_comics'
        download_xkcd_comics(output_folder)
```

6. **Write a program to read an image, Colouring and cropping image using Pillow module.**

```
        from PIL import Image, ImageEnhance
        def read_and_process_image(input_path, output_path):

            # Open the image file
            image = Image.open(input_path)

            # Apply color changes (enhance the color by a factor of 1.5)
            enhancer = ImageEnhance.Color(image)
            enhanced_image = enhancer.enhance(1.5)

            # Crop the image (left, top, right, bottom)
            cropped_image = enhanced_image.crop((50, 50, 300, 300))

            # Save the processed image
            cropped_image.save(output_path)

            # Display the original and processed images
            image.show(title='Original Image')
            cropped_image.show(title='Processed Image')

        # Example usage
        input_image_path = 'path/to/your/input/image.jpg'
        output_image_path = 'path/to/your/output/processed_image.jpg'
        read_and_process_image(input_image_path, output_image_path)
```