4. write a python program that reads a file containing a list of passwords, One per line. it checks each password to see if its meets certain requirements(eg. at least 8 characters, contains both uppercase and lowercase letters) and at least one number and one special character . passwords that satisfy the requirements Should be printed by the program.

```
import re

def is_valid_password(password):
# Password should be at least 8 characters long
    if len(password) < 8:
        return False

    # Password should contain both uppercase and lowercase letters
    if not re.search(r"[a-z]", password) or not re.search(r"[A-Z]", password):
        return False

    # Password should contain at least one digit
    if not re.search(r"\d", password):
        return False

     # Password should contain at least one special character
    if not re.search(r"[!@#$%^&*()-_=+{}|;:'\",.<>/?]", password):
        return False

    # If all requirements are met, the password is valid
    return True

def main():
    file_path = input("Enter the path to the file containing passwords: ")

    try:
        with open(file_path, 'r') as file:
            passwords = file.read().splitlines()

        for password in passwords:
            if is_valid_password(password):
                print("Valid password:", password)

    except FileNotFoundError:
        print("File not found. Please make sure the file exists.")

if __name__ == "__main__":
    main()
```

Input for the Program:

1. You need to create a password.txt file that contains list of password
   Password123!
   SecurePass
   weak
   StrongP@ssw0rd
   simple
   P@ssw0rd
   12345678
   VerySecure!Password

2. Then You need to enter the file path in order to execute the program successfully.

Explanation of the Program :

1. **Import Necessary Module:**
   - `import re`: This line imports a module called `re` that helps with pattern matching using regular expressions.

2. **Define a Function to Check Passwords:**
   - `is_valid_password(password)`: This function takes a password as input and checks if it meets certain criteria.

3. **Password Criteria:**
   - The password must be at least 8 characters long.
   - It should contain both uppercase and lowercase letters.
   - It must have at least one digit (number).
   - It should contain at least one special character (like !, @, #, etc.).

4. **Main Function:**
   - `main()`: This is the main part of the program where the user is prompted to enter the path to a file containing passwords.

5. **Try-Except Block:**
   - `try:` and `except FileNotFoundError:`: This block tries to open the specified file and read its contents. If the file is not found, it prints an error message.

6. **Read Passwords from File:**
   - `passwords = file.read().splitlines()`: Reads the contents of the file, splits it into lines, and stores each line (password) in a list called `passwords`.

7. **Loop Over Passwords:**
   - `for password in passwords:`: This loop goes through each password in the list.

8. **Check and Print Valid Passwords:**
   - `if is_valid_password(password):`: Checks if the current password is valid using the `is_valid_password` function.
   - If valid, it prints the password with the message "Valid password:".

9. **Example Usage:**
   - The program is designed to be run from the command line. When you run it, it asks you to provide the path to a file containing passwords.

10. **Example Input File (`passwords.txt`):**
    - An example file is provided (`passwords.txt`) with different passwords, some meeting the requirements and some not.

11. **Output:**
    - The program outputs valid passwords, helping you identify which passwords meet the specified security criteria.

The program's purpose is to assist in identifying strong passwords based on specific security requirements outlined in the code. You can adapt this program to check passwords in a file against your own security criteria.