This is a Python script that provides an interactive command-line interface to encrypt and decrypt messages using RSA encryption. It allows the user to choose between different options, including sending an encrypted message, decrypting a message, and generating new public and private keys. The script uses the **rsa** package to perform encryption and decryption operations.

Here is a brief overview of the main components of the script:

1. The script starts by importing the necessary modules, including **rsa**, **os**, and **msvcrt**.
2. The script then loads the public and private keys from the files "public.pem" and "private.pem", respectively, using the **rsa.PublicKey.load\_pkcs1()** and **rsa.PrivateKey.load\_pkcs1()** functions.
3. The script defines a function **press\_any\_key()** that waits for the user to press any key before continuing.
4. The script enters a while loop that displays a main menu and prompts the user to choose an option. The available options are:
   * Send an encrypted message
   * Decrypt a message
   * Generate new keys
   * Exit the script
5. If the user chooses the "Send an encrypted message" option, the script prompts the user to enter the recipient's name and the message to be encrypted. Depending on the recipient's name, the script encrypts the message using the appropriate public key and saves the encrypted message to a file. The script then calls the **press\_any\_key()** function to wait for the user to press any key before returning to the main menu.
6. If the user chooses the "Decrypt a message" option, the script prompts the user to enter the filename of the encrypted message to be decrypted. If the file exists, the script reads the contents of the file, decrypts the message using the private key, and displays the decrypted message to the user. The script then calls the **press\_any\_key()** function to wait for the user to press any key before returning to the main menu.
7. If the user chooses the "Generate new keys" option, the script generates new public and private keys using the **rsa.newkeys()** function and saves them to the files "public.pem" and "private.pem", respectively. The script then displays the new keys to the user and waits for the user to press any key before returning to the main menu.
8. If the user chooses the "Exit" option, the script exits the while loop and terminates.