

This is the working pseudocode solution to the program.

```
# Source Code File: Part 2: Symmetric Decryption using a Salsa20 Stream Cipher
# with Poly1305 Message Authentication Code
# Name: part2.py
# Author: <students name>

IMPORT SecretBox FROM nacl.secret
IMPORT CryptoError FROM nacl.exceptions
IMPORT sys

# Define a function for decryption
DEFINE FUNCTION decrypt_salsa20(ciphertext_path, key_path):
    """Decrypt a message encrypted with Salsa20 using a symmetric key."""
    TRY:
        # Open and read the ciphertext and key files
        OPEN ciphertext_path AS ciphertext_file, READ AS ciphertext
        OPEN key_path AS key_file, READ AS key

        # Check if the key length matches the required size
        IF LENGTH OF key IS NOT EQUAL TO SecretBox.KEY_SIZE:
            PRINT "Error: The key size must be exactly 32 bytes for Salsa20 encryption."
            EXIT PROGRAM WITH CODE 1

        # Create a SecretBox object and decrypt the ciphertext
        SET box TO SecretBox(key)
        SET plaintext TO RESULT OF box.decrypt(ciphertext)
        RETURN plaintext AS DECODED ASCII STRING

    EXCEPT FileNotFoundError AS error:
        RETURN "Error: " + error
    EXCEPT CryptoError:
        RETURN "Decryption failed. The ciphertext or key might be corrupted."
    EXCEPT UnicodeDecodeError:
        RETURN "Decryption succeeded, but the plaintext is not valid ASCII."

# Main program execution
IF SCRIPT IS RUN DIRECTLY:
    # Check if the correct number of arguments is provided
    IF NUMBER OF ARGUMENTS IS NOT 3:
        PRINT "Usage: python3 sym_decrypter.py <ciphertext_file> <key_file>"
        EXIT PROGRAM WITH CODE 1

    # Retrieve file paths from command-line arguments
    SET ciphertext_file TO ARGUMENT 1
    SET key_file TO ARGUMENT 2

    # Call the decryption function and display the message
    SET message TO RESULT OF decrypt_salsa20(ciphertext_file, key_file)
    PRINT message
```