This is the working pseudocode solution to the program.

```
# Source Code File:    Part 1: One-Time Pad Cryptography
# Name:                part1.py
# Author:              <students name>

IMPORT sys
IMPORT random FROM nacl.utils

# Define a function to reverse the one-time pad
DEFINE FUNCTION reverse_one_time_pad(ciphertext_path, otp_path):
    """Decrypt the ciphertext using the one-time pad."""
    TRY:
        # Open and read the ciphertext and OTP files
        OPEN ciphertext_path AS ciphertext_file, READ AS ciphertext
        OPEN otp_path AS otp_file, READ AS otp
    EXCEPT FileNotFoundError AS error:
        PRINT "Error:", error
        EXIT PROGRAM WITH CODE 1

    # Check if ciphertext and OTP are of equal length
    IF LENGTH OF ciphertext IS NOT EQUAL TO LENGTH OF otp:
        PRINT "Error: The ciphertext and OTP files must be of the same length."
        EXIT PROGRAM WITH CODE 1

    # Decrypt the ciphertext using XOR
    SET plaintext TO RESULT OF XORing EACH BYTE OF ciphertext WITH otp

    # Try to decode and print the plaintext
    TRY:
        PRINT plaintext AS ASCII
    EXCEPT UnicodeDecodeError:
        PRINT "Error: The decoded plaintext contains non-ASCII characters."

# Main program execution
IF SCRIPT IS RUN DIRECTLY:
    # Ensure two arguments (ciphertext and OTP paths) are provided
    IF NUMBER OF ARGUMENTS IS NOT 3:
        PRINT "Usage: python3 otp_decrypter.py <ciphertext_file> <otp_file>"
        EXIT PROGRAM WITH CODE 1

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

    # Call the reverse_one_time_pad function
    CALL reverse_one_time_pad(ciphertext_file, otp_file)
```