translate_message KEY translated = [] LETTERS = 'ABCDEFGHIJKLMNOPQRST...' key_index = 0 **✓** try def translate_message(key: str, message: str, mode: str) ->str:... key = key.upper() ∕ calls for symbol in message: input raise key.upper return message num = LETTERS.find(symbol.upper()) return ''.join(translated) default call __-_calls LETTERS.find if num != -1: num != -1 /(num == -1) translated.append(symbol) symbol.upper if mode == 'encrypt': calls (mode != 'encrypt') mode == 'encrypt' num += LETTERS.find(key[key_index]) if mode == 'decrypt': translated.append ,′calls mode == 'decrypt' num -= LETTERS.find(key[key_index]) LETTERS.find (mode != 'decrypt') calls (key_index != len(key)) num %= len(LETTERS) LETTERS.find __calls if symbol.isupper(): len _-- calls (not symbol.isupper()) symbol.isupper() if symbol.islower(): symbol.isupper translated.append(LETTERS[num]) symbol.islower() /calls `\calls (not symbol.islower()) translated.append calls key_index += 1 translated.append LETTERS.lower if key_index == len(key): key_index == len(key) calls $key_index = 0$ len

vigenere_cipher