MORSE\_CODE\_DICT = { 'A':'.-', 'B':'-...',

'C':'-.-.', 'D':'-..', 'E':'.',

'F':'..-.', 'G':'--.', 'H':'....',

'I':'..', 'J':'.---', 'K':'-.-',

'L':'.-..', 'M':'--', 'N':'-.',

'O':'---', 'P':'.--.', 'Q':'--.-',

'R':'.-.', 'S':'...', 'T':'-',

'U':'..-', 'V':'...-', 'W':'.--',

'X':'-..-', 'Y':'-.--', 'Z':'--..',

'1':'.----', '2':'..---', '3':'...--',

'4':'....-', '5':'.....', '6':'-....',

'7':'--...', '8':'---..', '9':'----.',

'0':'-----'}

# Function to encrypt the string

# according to the morse code chart

def encrypt(message):

cipher = ''

for letter in message:

if letter != ' ':

# Looks up the dictionary and adds the

# correspponding morse code

# along with a space to separate

# morse codes for different characters

cipher += MORSE\_CODE\_DICT[letter] + ' '

else:

# 1 space indicates different characters

# and 2 indicates different words

cipher += ' '

return cipher

# Function to decrypt the string

# from morse to english

def decrypt(message):

# extra space added at the end to access the

# last morse code

message += ' '

decipher = ''

citext = ''

for letter in message:

# checks for space

if (letter != ' '):

# counter to keep track of space

i = 0

# storing morse code of a single character

citext += letter

# in case of space

else:

# if i = 1 that indicates a new character

i += 1

# if i = 2 that indicates a new word

if i == 2 :

# adding space to separate words

decipher += ' '

else:

# accessing the keys using their values (reverse of encryption)

decipher += list(MORSE\_CODE\_DICT.keys())[list(MORSE\_CODE\_DICT

.values()).index(citext)]

citext = ''

return decipher

# Hard-coded driver function to run the program

def main():

message = input().strip().upper()

result = encrypt(message)

print (result.strip())

# Executes the main function

if \_\_name\_\_ == '\_\_main\_\_':

main()