🎯 Assignment Goal

Write a Python program from scratch that:

- **Encrypts or decrypts** a message.
- **Shifts each letter** by a key (number 1–25).
- Accepts only **valid input** for the key.
- Keeps looping until the user types `"quit"`.
- Leaves **spaces, numbers, and symbols unchanged**.

✏️ Step 1: Copy this starter code to your editor

```
# YOUR NAME(S)

# 7th Grade Computer Science

# TODAY'S DATE

# Caesar Cipher Program

# Caesar Cipher program will ask for a user input and the key they want to shift

# And encrypt/decrypt accordingly



message = input("Enter your message to encrypt: ")

shift = int(input("How many letters to shift by? "))

result = ""



for letter in message:

 if letter.islower():

    shifted = (ord(letter) - ord('a') + shift) % 26

    result += chr(shifted + ord('a'))

 else:
```

```
    result += letter  # leave everything else (symbol, numbers) unchanged


print("Encrypted message:", result)
```

🧪 What Does This Code Do?

- `ord(letter)` turns a character like `'a'` into a number (e.g., 97)
- `chr(number)` turns a number back into a letter
- `% 26` makes sure we "wrap around" the alphabet if we go past `'z'`
- Right now, it only works for lowercase letters

✏️ Step 2:Your Turn—Extend Your Program!

Now, extend your code:

Handle uppercase letters (`letter.isupper()`):

- Add another if statement using .isupper()
- Hint: uppercase letters use `'A'` instead of `'a'`
- Example: `"Hello"` with shift 3 → `"Khoor"`

Ignore numbers and symbols:

- The else already leaves these characters unchanged. Make sure to not delete it!!

Test Cases to try:

- `"Hello World!"`, shift 3 → `"Khoor Zruog!"`
- `"abc XYZ 123"`, shift 2 → `"cde ZAB 123"`

✏️ Step 3: Add User Option to Encrypt or Decrypt (5 min)

Change your program to ask the user if they want to encrypt or decrypt:

```
option = input("Type 'encrypt' or 'decrypt': ").lower()

message = input("Enter your message: ")

shift = int(input("Enter shift amount (1–25): "))

result = ""
```

✏️ Step 4: Wrap your encryption code with an "if" statement

Put all your encryption code inside an if-statement that checks if the user chose `"encrypt"`

✏️ Step 5: Code the Decryption Logic

Use similar logic as encryption, but shift **backwards (`- shift`)**. Add an `else` statement:

✏️ Step 6: Test Your Decryption (10–15 min)

Check your work carefully:

Decrypt messages such as:

- `"jgnnq"`, shift `2` → `"hello"`
- `"Udymts!"` shift `5` → `"Python!"`

✏️ Step 7: Keep your program running until user types `"quit"`

Put your main code inside a `while` loop

At the top of your main code, add:

```
while True:
```

And indent all your existing code

✏️ Step 8: Get User Option (Encrypt/Decrypt/Quit)

Change your program to ask the user if they want to encrypt or decrypt or quit:

```
option = input("Type 'encrypt', 'decrypt', or 'quit': ").lower()


 if option == "quit":
    print("Goodbye!")
    break
```

```
if option != "encrypt" and option != "decrypt":
    print("Invalid option. Please type 'encrypt', 'decrypt', or 'quit'.")
    continue
```

✏️ Step 9: Validate the shift key (accept only 1–25)

Add this if statement after user enters the key

```
if shift < 1 or shift > 25:

    print("Please enter a number between 1 and 25.")

    continue
```

🧪 Final Test Your Program

Carefully test your program. Be sure to test:

1. Valid Encryption/Decryption:

Option	Key	Message	Correct Result
Encrypt	3	hello	khoor
Decrypt	3	khoor	hello
Encrypt	5	Hello World!	Mjqqt Btwqi!
Decrypt	5	Mjqqt Btwqi!	Hello World!

2. Invalid Shift Keys:

Enter numbers like `0`, `-5`, `30`. Your program should prompt again.

3. Invalid Options:

Try typing something besides `"encrypt"`, `"decrypt"`, or `"quit"`. Your program should display an error and prompt again.

4. Symbols & Numbers:

Make sure messages like `"Hello! 123"` are encrypted or decrypted correctly (symbols and numbers remain unchanged).

-