Q1. Modify your greeting program so that if the user does not enter a name (i.e. the just press enter), the program responds "Hello, Stranger!". Otherwise it should print a greeting with their name as before.

Ans:

```
def greeting_program():
    name = input("Please enter your name: ").strip()
    if name:
        print(f"Hello, {name}!")
    else:
        print("Hello, Stranger!")

greeting_program()
```

Output:

Please enter your name: Ghanshyam

Hello, Ghanshyam!

Q2. Write a program that simulates the way in which a user might choose a password. The program should prompt for a new password, and then prompt again. If the two passwords entered are the same the program should say "Password Set" or similar, otherwise it should report an error.

Ans:

```
user_password=input("Please enter a new password: ")
conform_password=input("Please conform your password: ")

if user_password==conform_password:
    print("Password Set")
```

```
else:
   print("Error: Password donot match.")
```

Output:

Please enter a new password: Rolpa@123

Please confirm your password: Rolpa@123

Password Set

Q3. Modify your previous program so that the password must be between 8 and 12 characters (inclusive) long.

Ans:

```
password= input("Enter a password: ")
if len(password)<8 or len(password)>12:
   print("Error: Password must be between 8 and 12 characters long.")

else:
   print("Password set.")
```

Output:

Enter a password: Rolpa

Error: Password must be between 8 and 12 characters long.

Q4. Modify your program again so that the chosen password cannot be one of a list of common passwords, defined thus:

```
BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']'''
```

Ans:

```
BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
user_password = input("Please enter a new password: ")
confirm_password = input("Please confirm your password: ")
if user_password == confirm_password:
    if len(user_password) >= 8 and len(user_password) <= 12:
        if user_password not in BAD_PASSWORDS:
            print("Password Set")
        else:
            print("Error: Password is too common.")
    else:
        print("Error: Password must be between 8 and 12 characters.")

else:
    print("Error: Passwords do not match.")
```

Output:

Please enter a new password: password

Please confirm your password: password

Error: Password is too common.

Q5. Modify your program a final time so that it executes until the user successfully chooses a password. That is, if the password chosen fails any of the checks, the program should return to asking for the password the first time.

Ans:

```
BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
```

```
while True:
    password = input("Choose a password: ")

    if password in BAD_PASSWORDS:
        print("That password is too common. Please choose a more secure password.")
    elif not password:
        print("Password cannot be empty. Please try again.")
    else:
        print("Password accepted!")
        break
```

Output:

Choose a password: hello

That password is too common. Please choose a more secure password.

Choose a password:

Q6. Write a program that displays the "Seven Times Table". That is, the result of multiplying 7 by every number from 0 to 12 inclusive. The output might start:

```
0 x 7 = 0
1 x 7 = 7
2 x 7 = 14
and so on.'''
```

Ans:

```
for i in range(13):
    print(f"{i} x 7 = {i * 7}")
```

Output:

0 x 7 = 0

1 x 7 = 7

2 x 7 = 14

3 x 7 = 21

4 x 7 = 28

5 x 7 = 35

6 x 7 = 42

7 x 7 = 49

8 x 7 = 56

9 x 7 = 63

10 x 7 = 70

11 x 7 = 77

12 x 7 = 84

Q7. Modify your "Times Table" program so that the user enters the number of the table they require. This number should be between 0 and 12 inclusive.

Ans:

```
while True:
    try:
        number = int(input("Enter the number for the times table (0-12): "))
        if 0 <= number <= 12:
            break
        else:
            print("Please enter a number between 0 and 12.")
    except ValueError:
        print("Invalid input. Please enter a valid number.")


print(f"\n{number} Times Table:")
```

```
for i in range(13):
    print(f"{i} x {number} = {i * number}")
```

Output:

Enter the number for the times table (0-12): 5

5 Times Table:

0 x 5 = 0

1 x 5 = 5

2 x 5 = 10

3 x 5 = 15

4 x 5 = 20

5 x 5 = 25

6 x 5 = 30

7 x 5 = 35

8 x 5 = 40

9 x 5 = 45

10 x 5 = 50

11 x 5 = 55

12 x 5 = 60

Q8. Modify the "Times Table" again so that the user still enters the number of the table, but if this number is negative the table is printed backwards. So entering "-7" would produce the Seven Times Table starting at "12 times" down to "0 times".

Ans:

```
while True:
    try:
```

```
        number = int(input("Enter the number for the times table (positive for normal, negative for reverse): "))
        if -12 <= number <= 12:
            break
        else:
            print("Please enter a number between -12 and 12.")
    except ValueError:
        print("Invalid input. Please enter a valid number.")

table_number = abs(number)
if number < 0:
    print(f"\n{table_number} Times Table (Reverse):")
    for i in range(12, -1, -1):
        print(f"{i} x {table_number} = {i * table_number}")
else:
    print(f"\n{table_number} Times Table:")
    for i in range(13):
        print(f"{i} x {table_number} = {i * table_number}")
```

Output:

Enter the number for the times table (positive for normal, negative for reverse): -7

7 Times Table (Reverse):

12 x 7 = 84

11 x 7 = 77

10 x 7 = 70

9 x 7 = 63

8 x 7 = 56

7 x 7 = 49

6 x 7 = 42

5 x 7 = 35

4 x 7 = 28

3 x 7 = 21

2 x 7 = 14

1 x 7 = 7

0 x 7 = 0