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

name=str(input("Enter your name :"))
if name== "" :
    print('"Hello,Stranger!"')
else:
    print("Hello,",name,"Good to meet you!")
```

Enter your name : Susha

Hello, Susha Good to meet you!

```python
# 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.


psw1=(input("Enter the password : "))
psw2=(input("Enter the password again : "))
if psw1==psw2:
    print("Password Set")
if psw1!= psw2:
    print("Error occured")
```

Enter the password : Sit
Enter the password again : Sit

Password Set

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


psw = input("Enter the password: ")
if len(psw)<8 or len(psw)>12:
    print("Password must be between 8 to 12 characters long.")
else:
    check_psw = input("Confirm the password: ")

    if psw == check_psw:
        print("Password Set")
    else:
        print("Sorry! The password did not match")
```

Enter the password: sundown12345
Confirm the password: sundown12345

Password Set

```python
# 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']

bad_passwords= ['password', 'letmein', 'sesame', 'hello',
'justinbieber']

psw=input("Enter a password :")
if psw in bad_passwords :
    print("The given password could not be set")
else:
    print("password set")
```

```
Enter a password : password

The given password could not be set
```

```python
#  Write a program that displays the "Seven Times Table". That is, the
result of multiplying 7 by every number from 0 to 12 inclusive

print("Multiplication table of 7 :")
for i in range(13):
    print(7,"*",i,"=",7*i)
```

```
Multiplication table of 7 :
7 * 0 = 0
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
7 * 11 = 77
7 * 12 = 84
```

```python
# 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.

number=int(input("Enter the number :"))
print("The multiplication table of",number)
if number>=0 and number<=12 :
    for i in range(11):
        print(number,"*",i,"=",number*i)
```

```
else:
    print("number not within the range")
```

```
Enter the number : 9

The multiplication table of 9
9 * 0 = 0
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
```

```
# 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"

number=int(input("Enter the number :"))
print("The multiplication table of",number)
if number>=0 and number<=12 :
    for i in range(11):
        print(number,"*",i,"=",number*i)
elif number<=0 and number<=12 :
    for i in range(12,1,-1):
        print(number,"*",i,"=",number*i)
else:
    print("number not within the range")
```

```
Enter the number : -7

The multiplication table of -7
-7 * 12 = -84
-7 * 11 = -77
-7 * 10 = -70
-7 * 9 = -63
-7 * 8 = -56
-7 * 7 = -49
-7 * 6 = -42
-7 * 5 = -35
-7 * 4 = -28
-7 * 3 = -21
-7 * 2 = -14
```