In [3]:

```
#Question number 1
#Modify your greeting program so that if the user does not enter a name (i.e. they just press enter), the progra

# Ask the user to enter their name
name = input("Enter your name: ")

# Check if the 'name' variable contains any input
if name:
    # If 'name' is not empty, greet the user by their name
    print(f"Hello, {name}")
else:
    # If 'name' is empty (the user didn't type anything), greet the user as "Stranger"
    print(f"Hello, Stranger")
```

```
Hello, Stranger
```

In [4]:

```
#Question number 2
#Write a program that simulates the way in which a user might choose a password.The program should prompt for a


# Ask the user to enter their desired password for the first time
first_pw = input("Enter the password of choice: ")

# Ask the user to enter their desired password again for confirmation
second_pw = input("Enter the password again: ")

# Check if both passwords are the same
if first_pw == second_pw:
    # If the passwords match, print a message confirming the password is set
    print("Password set")
else:
    # If the passwords do not match, print an error message and ask the user to try again
    print("Passwords do not match, Try again..")
```

```
Password set
```

In [5]:

```
#Question number 3
#Modify your previous program so that the password must be between 8 and 12 characters (inclusive) long.

# Ask the user to enter their desired password for the first time
first_pw = input("Enter the password of choice between 8-12 characters: ")


if len(first_pw)>=8 and len(first_pw)<=12:
    # Ask the user to enter their desired password again for confirmation
    second_pw= input("Enter your password again: ")

    # Check if both passwords are the same
    if first_pw == second_pw:
        # If the passwords match, print a message confirming the password is set
        print("Password set")
    else:
        # If the passwords do not match, print an error message and ask the user to try again
        print("Passwords do not match, Try again..")

else:
    print("Your password does not match the requirements")
```

```
Your password does not match the requirements
```

In [2]:

```
#Question number 4
#Modify your program again so that the chosen password cannot be one of a list of common passwords, defined thu

BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
first_pw= input("Enter the password of choice between 8-12 characters: ")

if first_pw in BAD_PASSWORDS:
    print("You can not choose passwords among these ['password', 'letmein', 'sesame', 'hello', 'justinbieber']"

else:
    if len(first_pw)>=8 and len(first_pw)<=12:
        # Ask the user to enter their desired password again for confirmation
        second_pw= input("Enter your password again: ")

        # Check if both passwords are the same
        if first_pw == second_pw:
            # If the passwords match, print a message confirming the password is set
            print("Password set")
        else:
            # If the passwords do not match, print an error message and ask the user to try again
            print("Passwords do not match, Try again..")

    else:
        print("Your password does not match the requirements")
```

```
You can not choose passwords among these ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
```

In [6]:

```
#Question number 5
# Modify your program a final time so that it executes until the user successfully chooses a password. That is,

#List of common bad passwords
BAD_PASSWORDS = ['password', 'letmein', 'sesame', 'hello', 'justinbieber']

while True:
    # Prompt the user to enter their password of choice
    first_pw = input("Enter the password of choice between 8-12 characters: ")

    #checks if the password is among the list or not
    if first_pw in BAD_PASSWORDS:
        print("Password cannot be among these bad passwords ['password', 'letmein', 'sesame', 'hello', 'justinb

    else:
        #checks if the password entered meets the desirable length
        if len(first_pw)>=8 and len(first_pw)<=12:
            # Ask the user to enter their desired password again for confirmation
            second_pw= input("Enter your password again: ")

            # Check if both passwords are the same
            if first_pw == second_pw:
                # If the passwords match, print a message confirming the password is set
                print("Password set")
                break

            else:
                # If the passwords do not match, print an error message and ask the user to try again
                print("Passwords do not match, Try again..")

        else:
            print("Your password does not match the requirements")
```

```
Password cannot be among these bad passwords ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
Password cannot be among these bad passwords ['password', 'letmein', 'sesame', 'hello', 'justinbieber']
Password set
```

In [7]:

```
# #Question number 6
# #Write a program that displays the "Seven Times Table". That is, the result of multiplying 7 by every number 
# 0 x 7 = 0
# 1 x 7 = 7
# 2 x 7 = 14
# and so on.


#initialize i to 0
i=0

#start a loop that runs as long as i is less or equal to 12
while i<=12:
    #print the multiplication table of 7
    print(f"{i}*{7}= {i*7}")
#increament i by 1
    i+=1
```

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

In [8]:

```
#Question number 7
#Modify your "Times Table" program so that the user enters the number of the table they require. This number sh

# Prompt the user to enter a number
number = int(input("Enter a number to generate a number: "))

# start aoop from 0 to 12 (inclusive)
for i in range(13):
    #print the multiplication table of the number the user enters as output.
    print(f"{i} * {number} = {i * number}")
```

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

In [9]:

```python
#Question nhumber 8
#Modify the "Times Table" again so that the user still enters the number of the table, but if this number is neg


# Prompt the user to enter a number
number = int(input("Enter a number to generate a number:"))

# Check if the number is negative
if number < 0:
    # Loop from 12 down to 0 (inclusive) in reverse order
    for i in range(12, -1, -1):
        # Print the multiplication table for the negative number
        print(f"{i}*{number}={i*number}")
else:
    # Loop from 0 to 12 (inclusive)
    for i in range(13):
        # Print the multiplication table for the positive number
        print(f"{i}*{number}={i*number}")
```

```
12*-4=-48
11*-4=-44
10*-4=-40
9*-4=-36
8*-4=-32
7*-4=-28
6*-4=-24
5*-4=-20
4*-4=-16
3*-4=-12
2*-4=-8
1*-4=-4
0*-4=0
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js