In [3]:

```python
# Guess a number is a game that prompts a player to guess a number between 0 and 9, which
# is randomly generated by the system. When the input given by the user matches the
# number generated by the system then the user wins. The game should go as follows:

# Guess the number: 5
# Sorry, try again
# Guess the number: 3
# Sorry, try again
# Guess the number: 8
# You got it right! Congo!


import random
def guess_number():
    number=random.randint(0,9)

    while True:
        user=(eval(input("guess the number (between 0 t0 9):")))

        if user==number:
            print("you got it right!congratulations!")
            break
        else:
            print("sorry, try again")

guess_number()
```

```
guess the number (between 0 t0 9):5
sorry, try again
guess the number (between 0 t0 9):3
sorry, try again
guess the number (between 0 t0 9):8
sorry, try again
guess the number (between 0 t0 9):7
sorry, try again
guess the number (between 0 t0 9):9
sorry, try again
guess the number (between 0 t0 9):1
sorry, try again
guess the number (between 0 t0 9):4
you got it right!congratulations!
```

In [5]:

```python
# Make an improvement to the Guess a number game. Guide the user where they are
# standing and limit the number of attempts to 3. For example, the game should go like this:

# Guess the number: 4
# Too low
# Guess the number: 9
# Too high
# Guess the number: 5
# Sorry you Lost!

import random
def guess_number():
    number=random.randint(0,9)

    attempts=3

    while attempts>0:
        user=(eval(input("guess the number (between 0 t0 9):")))

        if user==number:
            print("congratulations! you got it right!")
            break
        elif user<number:
            print("too low")
        else:
            print("too high")

        attempts-=1
        print("attempts left {}:".format(attempts))
    if attempts==0:
        print("sorry you lost:")

guess_number()
```

```
guess the number (between 0 t0 9):6
too low
attempts left 2:
guess the number (between 0 t0 9):9
too high
attempts left 1:
guess the number (between 0 t0 9):10
too high
attempts left 0:
sorry you lost:
```

In [28]:

```python
# Let us make the above game a little more interesting by converting it into a gamblingproblem.
# Suppose that a player starts with Rs. 1,000. If a player can guess the number in his
# first chance, then he will be given a prize of Rs. 5,000, if he requires 2 attempts then he will
# get a prize of Rs.1,000. If he loses then he will lose Rs. 500. For example the game should go like this:

# You have a cash of Rs. 1,000 with you...
# Guess the number: 8
# Too high
# Guess the number: 3
# You have just won Rs. 1,000

# Your balance: Rs. 2000

import random
def guess_number():
    balance="1000"
    print("cash 1000 with you:")
    number=random.randint(1,10)
    for attempts in range(1,3):
        guess=eval(input("guess the number:"))
        if guess_number==number:
            if attempts==1:
                prize=5000
                print("congratutaltions! you guess the number in your first attempt:")
            elif attempt==2:
                prize=1000
                print("congratutaltions! you guess the number in your second attempt:")
            break
        elif guess<number:
            print("too high")
        else:
            print("too low")
    else:
        prize=-500
        print("sorry, you couldnot guess the number {}:".format(number))
        balance=prize
        print("you have just {}:".format(prize))
        print("you balance {}:".format(balance))

guess_number()
```

```
cash 1000 with you:
guess the number:500
too low
guess the number:600
too low
sorry, you couldnot guess the number 8:
you have just -500:
you balance -500:
```

In [40]:

```python
# Suppose that a player wants to play a game which requires him Rs. 1,000 to start.
# If thecurrent balance in his account is less than Rs. 1,000 he needs to withdraw the extra money
# from his e-wallet.

# Note that if the sum of money in his courrent account and the amount withdrawn is greater than
# or equal to Rs. 1,000 then he can start playing the game. However if the sum is less than Rs.
# 1,000 then the program should keep displaying the user the message “You still do not have
# enough money to start playing.” and keep prompting the user to withdraw money unless it
# crosses Rs. 1,000. Once ready, i.e. if his current account balance crosses Rs. 1,000, it will display
# a message “Now, you are ready to play the game.” Your program should also display the account
# balance and the current amount in the e-wallet.

#(consider: initial account balance is Rs. 200 and money in the e-wallet is Rs. 5,000)

#(Do further improvement by checking if the e-wallet balance becomes NIL, etc.)




def play_game(account_balance, e_wallet_balance):
    while account_balance + e_wallet_balance < 1000:
        print("You still do not have enough money to start playing:")

        withdraw_amount = int(input("Enter the amount you want to withdraw: "))

        if e_wallet_balance >= withdraw_amount:
            account_balance += withdraw_amount
            e_wallet_balance -= withdraw_amount
        else:
            print("Insufficient funds in e-wallet. Please try again.")

        print("Account Balance: Rs. {account_balance}")
        print("E-Wallet Balance: Rs. {e_wallet_balance}")

    print("Now, you are ready to play the game.")
    print("Account Balance: Rs. {}:".format(account_balance))
    print("E-Wallet Balance: Rs. {}:".format(e_wallet_balance))

initial_account_balance = 200
initial_e_wallet_balance = 5000

play_game(initial_account_balance, initial_e_wallet_balance)
```

```
Now, you are ready to play the game.
Account Balance: Rs. 200:
E-Wallet Balance: Rs. 5000:
```

In [ ]: