**Driver.py**

from Bully import Bully

#Dummy Processes

process\_count = int(input("Enter Number of Processes"))

bully = Bully(process\_count)

bully.intiailzeProcesses()

state = True

while state:

    print("1. Initalize the process\n2. Bring Down process\n3. Activate Process\n4. Exit \n 5. Current Coordinator\n")

    choice = int(input())

    if(choice==1):

        bully.intiailzeProcesses()

    elif(choice==2):

        crash\_id = int(input("Enter the process you want to crash"))

        bully.crash(crash\_id)

    elif(choice==3):

        process\_id = int(input("Enter the process you want to start"))

        bully.start(process\_id)

    elif(choice==4):

        state=False

        print("Exiting the program")

    elif(choice==5):

        bully.coordinator()

    else:

        print("Invalid Input")

**Bully.py**

from statistics import mode

class Process:

    def \_\_init\_\_(self, process\_id, total\_count):

        self.process\_id = process\_id

        self.total\_count = total\_count

        self.leader\_id = -1

        self.is\_active = True

    def crash(self):

        self.is\_active = False

    def start(self):

        self.is\_active = True

    def is\_leader(self):

        if self.process\_id == self.leader\_id:

            return True

        return False

    def set\_leader(self, leader):

        self.leader\_id = leader

    def get\_leader(self):

        return self.leader\_id

    def sendRequest(self, toProcess):

        print(f"Sending request to process {toProcess.process\_id} from {self.process\_id}")

        if(toProcess.reciveRequest(self.process\_id)):

            print(f"Ok recived from {toProcess.process\_id}")

            self.set\_leader(toProcess.process\_id)

        else:

            print(f"No response from {toProcess.process\_id}")

    def reciveRequest(self, fromProcess):

        if(self.is\_active):

            print(f"Recived request from process {fromProcess}.")

            return self.recivedMessage()

        return False

    def recivedMessage(self):

            return True;

class Bully:

    def \_\_init\_\_(self, total\_count):

        self.processes = []

        self.total\_count = total\_count

        # self.leader = None

    def intiailzeProcesses(self):

        self.processes = []

        for i in range(self.total\_count):

            self.processes.append(Process(i, total\_count = self.total\_count))

        self.elect\_leader()

        self.coordinator()

    def elect\_leader(self, current=0):

        for i in range(current, self.total\_count):

            if self.processes[i].is\_active:

                # [self.processes[i].sendRequest(self.processes[j]) for j in range(i, self.total\_count)]

                for j in range(i+1, self.total\_count):

                    if(self.processes[j].is\_active):

                        self.processes[i].sendRequest(self.processes[j])

                    elif(not self.processes[j].is\_active and i+1==self.total\_count-1):

                        self.processes[i].sendRequest(self.processes[i])

                if self.processes[i].get\_leader()==-1:

                    self.processes[i].sendRequest(self.processes[i])

                # if(i==self.total\_count-1):

                #     self.processes[i].sendRequest(self.processes[i])

    def crash(self, crash\_id):

        if(crash\_id<self.total\_count and crash\_id>=0):

            self.processes[crash\_id].crash()

            # print(f"Process id {Process.process\_id} crashed.")

            if(self.processes[crash\_id].is\_leader()):

                print("Leader process Down.\n Initaling the leader lookout.")

                self.elect\_leader(0)

    def start(self, process\_id):

        if(self.processes[process\_id].is\_active):

            print("Process already active")

        else:

            self.processes[process\_id].start()

            self.elect\_leader()

            # if(self.processes[process\_id].is\_active):

            #     if process\_id>self.processes[self.leader].get\_leader():

                    # self.elect\_leader(self.leader)

    def coordinator(self):

        leader = []

        for p in self.processes:

            if p.is\_active:

                print(p.get\_leader())

                leader.append(p.get\_leader())

        self.leader = mode(leader)

Run on terminal: py Driver.py