# Python3 program for implementation

# of FCFS scheduling

# Function to find the waiting

# time for all processes

**def** findWaitingTime(processes, n,

                    bt, wt):

    # waiting time for

    # first process is 0

    wt[0] **=** 0

    # calculating waiting time

**for** i **in** range(1, n ):

        wt[i] **=** bt[i **-** 1] **+** wt[i **-** 1]

# Function to calculate turn

# around time

**def** findTurnAroundTime(processes, n,

                       bt, wt, tat):

    # calculating turnaround

    # time by adding bt[i] + wt[i]

**for** i **in** range(n):

        tat[i] **=** bt[i] **+** wt[i]

# Function to calculate

# average time

**def** findavgTime( processes, n, bt):

    wt **=** [0] **\*** n

    tat **=** [0] **\*** n

    total\_wt **=** 0

    total\_tat **=** 0

    # Function to find waiting

    # time of all processes

    findWaitingTime(processes, n, bt, wt)

    # Function to find turn around

    # time for all processes

    findTurnAroundTime(processes, n,

                       bt, wt, tat)

    # Display processes along

    # with all details

    print( "Processes Burst time " **+**

                  " Waiting time " **+**

                " Turn around time")

    # Calculate total waiting time

    # and total turn around time

**for** i **in** range(n):

        total\_wt **=** total\_wt **+** wt[i]

        total\_tat **=** total\_tat **+** tat[i]

**print**(" " **+** str(i **+** 1) **+** "\t\t" **+**

                    str(bt[i]) **+** "\t " **+**

                    str(wt[i]) **+** "\t\t " **+**

                    str(tat[i]))

    print( "Average waiting time = "**+**

                   str(total\_wt **/** n))

    print("Average turn around time = "**+**

                     str(total\_tat **/** n))

# Driver code

**if** \_\_name\_\_ **==**"\_\_main\_\_":

    # process id's

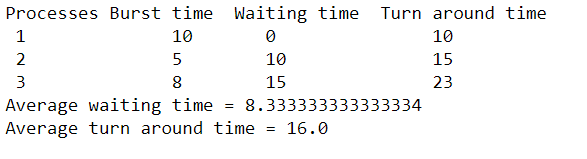
    processes **=** [ 1, 2, 3]

    n **=** len(processes)

    # Burst time of all processes

    burst\_time **=** [10, 5, 8]

    findavgTime(processes, n, burst\_time)



# Python3 program to implement Shortest Remaining Time First

# Shortest Remaining Time First (SRTF)

# Function to find the waiting time

# for all processes

**def** findWaitingTime(processes, n, wt):

    rt **=** [0] **\*** n

    # Copy the burst time into rt[]

**for** i **in** range(n):

        rt[i] **=** processes[i][1]

    complete **=** 0

    t **=** 0

    minm **=** 999999999

    short **=** 0

    check **=** False

    # Process until all processes gets

    # completed

**while** (complete !**=** n):

        # Find process with minimum remaining

        # time among the processes that

        # arrives till the current time`

**for** j **in** range(n):

**if** ((processes[j][2] <**=** t) **and**

                (rt[j] < minm) **and** rt[j] > 0):

                minm **=** rt[j]

                short **=** j

                check **=** True

**if** (check **==** False):

            t **+=** 1

**continue**

        # Reduce remaining time by one

        rt[short] **-=** 1

        # Update minimum

        minm **=** rt[short]

**if** (minm **==** 0):

            minm **=** 999999999

        # If a process gets completely

        # executed

**if** (rt[short] **==** 0):

            # Increment complete

            complete **+=** 1

            check **=** False

            # Find finish time of current

            # process

            fint **=** t **+** 1

            # Calculate waiting time

            wt[short] **=** (fint **-** proc[short][1] **-**

                                proc[short][2])

**if** (wt[short] < 0):

                wt[short] **=** 0

        # Increment time

        t **+=** 1

# Function to calculate turn around time

**def** findTurnAroundTime(processes, n, wt, tat):

    # Calculating turnaround time

**for** i **in** range(n):

        tat[i] **=** processes[i][1] **+** wt[i]

# Function to calculate average waiting

# and turn-around times.

**def** findavgTime(processes, n):

    wt **=** [0] **\*** n

    tat **=** [0] **\*** n

    # Function to find waiting time

    # of all processes

    findWaitingTime(processes, n, wt)

    # Function to find turn around time

    # for all processes

    findTurnAroundTime(processes, n, wt, tat)

    # Display processes along with all details

**print**("Processes    Burst Time     Waiting",

                    "Time     Turn-Around Time")

    total\_wt **=** 0

    total\_tat **=** 0

**for** i **in** range(n):

        total\_wt **=** total\_wt **+** wt[i]

        total\_tat **=** total\_tat **+** tat[i]

        print(" ", processes[i][0], "\t\t",

                   processes[i][1], "\t\t",

                   wt[i], "\t\t", tat[i])

**print**("\nAverage waiting time = %.5f "**%**(total\_wt **/**n) )

**print**("Average turn around time = ", total\_tat **/** n)

# Driver code

**if** \_\_name\_\_ **==**"\_\_main\_\_":

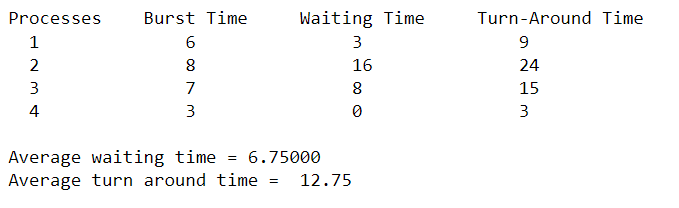
    # Process id's

    proc **=** [[1, 6, 1], [2, 8, 1],

            [3, 7, 2], [4, 3, 3]]

    n **=** 4

    findavgTime(proc, n)



# Python3 program for implementation of

# Priority Scheduling

# Function to find the waiting time

# for all processes

**def** findWaitingTime(processes, n, wt):

    wt[0] **=** 0

    # calculating waiting time

**for** i **in** range(1, n):

        wt[i] **=** processes[i **-** 1][1] **+** wt[i **-** 1]

# Function to calculate turn around time

**def** findTurnAroundTime(processes, n, wt, tat):

    # Calculating turnaround time by

    # adding bt[i] + wt[i]

**for** i **in** range(n):

        tat[i] **=** processes[i][1] **+** wt[i]

# Function to calculate average waiting

# and turn-around times.

**def** findavgTime(processes, n):

    wt **=** [0] **\*** n

    tat **=** [0] **\*** n

    # Function to find waiting time

    # of all processes

    findWaitingTime(processes, n, wt)

    # Function to find turn around time

    # for all processes

    findTurnAroundTime(processes, n, wt, tat)

    # Display processes along with all details

    print("\nProcesses    Burst Time    Waiting",

          "Time    Turn-Around Time")

    total\_wt **=** 0

    total\_tat **=** 0

**for** i **in** range(n):

        total\_wt **=** total\_wt **+** wt[i]

        total\_tat **=** total\_tat **+** tat[i]

**print**(" ", processes[i][0], "\t\t",

              processes[i][1], "\t\t",

              wt[i], "\t\t", tat[i])

    print("\nAverage waiting time = %.5f " **%** (total\_wt **/** n))

    print("Average turn around time = ", total\_tat **/** n)

**def** priorityScheduling(proc, n):

    # Sort processes by priority

    proc **=** sorted(proc, key**=lambda** proc: proc[2],

                  reverse**=**True)

    print("Order in which processes gets executed")

**for** i **in** proc:

        print(i[0], end**=**" ")

    findavgTime(proc, n)

# Driver code

**if** \_\_name\_\_ **==** "\_\_main\_\_":

    # Process id's

    proc **=** [[1, 10, 1],

            [2, 5, 0],

            [3, 8, 1]]

    n **=** 3

    priorityScheduling(proc, n)

