PRIORITY

import java.util.Scanner;

public class scheduling {

    public static void sortcolm(int arr[][], int start, int end, int colm) {

        for (int a = start; a < end - 1; a++) {

            int min = a;

            for (int b = a + 1; b < end; b++) {

                if (arr[b][colm] < arr[min][colm]) {

                    min = b;

                }

            }

            int temp1 = arr[a][0];

            arr[a][0] = arr[min][0];

            arr[min][0] = temp1;

            int temp2 = arr[a][1];

            arr[a][1] = arr[min][1];

            arr[min][1] = temp2;

            int temp3 = arr[a][2];

            arr[a][2] = arr[min][2];

            arr[min][2] = temp3;

        }

    }

    public static void main(String[] args) {

        int p\_no;

        float avgTAT = 0, avgWT = 0;

        int process[][] = new int[10][6];

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the no. of processes: ");

        p\_no = sc.nextInt();

        for (int i = 0; i < p\_no; i++) {

            System.out.print("Arrival Time: ");

            process[i][0] = sc.nextInt();

            System.out.print("Burst Time: ");

            process[i][1] = sc.nextInt();

            System.out.print("Priority: ");

            process[i][2] = sc.nextInt();

        }

        sortcolm(process, 0, p\_no, 0);

        int flag = 0, c\_ptr = process[0][1];

        for (int a = 0; a < p\_no - 1; a++) {

            int first = a + 1, last = 0;

            for (int b = a + 1; b < p\_no; b++) {

                if (process[b][0] <= c\_ptr) {

                    last = b;

                    flag = 1;

                } else {

                    break;

                }

            }

            if (flag == 1) {

                sortcolm(process, first, last + 1, 2);

                c\_ptr = c\_ptr + process[first][1];

            }

        }

        int temp = 0;

        for (int j = 0; j < p\_no; j++) {

            process[j][3] = temp + process[j][1];

            temp = process[j][3];

            process[j][4] = process[j][3] - process[j][0];

            avgTAT = avgTAT + process[j][4];

            process[j][5] = process[j][4] - process[j][1];

            avgWT = avgWT + process[j][5];

        }

        avgTAT = avgTAT / p\_no;

        avgWT = avgWT / p\_no;

        // 7 032 256 143 425 697 544 71010

        // print Priority(non pre-emptive) table

        System.out.println("\n\tPriority Scheduling");

        System.out.println("success");

        System.out.println("Arrival | Burst | Priority | Completion | TurnAround | Waiting");

        System.out.println("success");

        System.out.println("--------------------------------------------------------------");

        for (int k = 0; k < p\_no; k++) {

            System.out.println(process[k][0] + "\t| " + process[k][1] + "\t| " + process[k][2] + "\t | " + process[k][3]

                    + "\t\t| " + process[k][4] + "\t | " + process[k][5]);

        }

        System.out.println("--------------------------------------------------------------");

        System.out.println("Average Turn Around Time: " + avgTAT);

        System.out.println("Average Waiting Time: " + avgWT);

        sc.close();

    }

}