Process Scheduling 1 (scheduling1)

[Time Limit : 1 sec , Mem Limit : 32 MB]

Problem :

ซีพียู (CPU) คือหน่วยประมวลผลกลางของเครื่องคอมพิวเตอร์ซึ่งต้องทำงานต่างๆมากมาย โดยงานต่างๆที่ซีพียูต้องทำจะเกิดจากสิ่งที่เรียกว่า โปรเซส (Process) ในขณะใดขณะหนึ่งซีพียูจะสามารถทำงานได้เพียง 1 งานเท่านั้น ดังนั้นหากมีมากกว่า 1 โปรเซส ที่ต้องการเข้าไปทำงานในซีพียู ก็จะต้องมีอัลกอริทึมสำหรับการทำ Process Scheduling เพื่อเลือกลำดับว่าโปรเซสใดจะได้เข้าไปทำงานในซีพียูก่อน (หรือหลัง)

หนึ่งในอัลกอริทึมพื้นฐานคือ FCFS (First Come First Serve) โดยปกติแล้วจะมีแถวสำหรับให้โปรเซสต่อแถวเพื่อรอเข้าไปทำงานในซีพียู ซึ่งเรียกว่า Wait Queue ซึ่งหลักการก็คือโปรเซสใดที่เกิดก่อนก็จะได้เข้ามาต่อท้ายแถวใน Wait Queue ก่อน และเมื่อซีพียูว่าง โปรเซสที่อยู่หัวแถวก็จะออกจากแถวและเข้ามาทำงานในซีพียูจนกว่าจะเสร็จ จากนั้นซีพียูก็จะว่างอีกครั้งและเอาโปรเซสอันถัดไปที่อยู่หัวแถวเข้ามาทำงาน และเป็นเช่นนี้ไปเรื่อยๆ

ทั้งนี้สำหรับอัลกอริทึมใดๆย่อมมีการวัดประสิทธิภาพ ซึ่งสำหรับการทำ Process Scheduling นี้ก็จะมีวิธีวัดประสิทธิภาพจากการคำนวณ Average Waiting Time ซึ่งก็คือค่าเฉลี่ยของเวลาที่ทุกๆโปรเซสต้องรออยู่ใน Wait Queue จงเขียนโปรแกรมเพื่อรับข้อมูลของแต่ละโปรเซส ได้แก่เวลาเกิดและเวลาทั้งหมดที่คาดว่าโปรเซสนั้นจะใช้ในการทำงานในซีพียู และคำนวณ Average Waiting Time ออกมา

Input :

บรรทัดแรก ระบุจำนวนเต็ม T (1 <= T <= 5) แทนจำนวนชุดทดสอบย่อยทั้งหมด

สำหรับแต่ละชุดทดสอบย่อย :

บรรทัดแรก ระบุจำนวนเต็ม N (1 <= N <= 100,000) แทนจำนวนโปรเซสทั้งหมด

อีก N บรรทัด โดยบรรทัดที่ i (1 <= i <= N) ระบุจำนวนเต็ม P_i, Q_i (1 <= P_i <= 10^9; 1 <= Q_i <= 10,000) แทนเวลาเกิดและระยะเวลาที่จะใช้ในการทำงานของโปรเซสที่ i ตามลำดับ ข้อมูลเหล่านี้อาจจะไม่เรียงลำดับตามเวลาเกิด

บรรทัดสุดท้าย ระบุจำนวนเต็ม M (3 <= M <= 10) แทนจำนวนหลักของตัวเลขหลังจุดทศนิยม เป็นการระบุความละเอียดของคำตอบ

Output :

สำหรับแต่ละชุดทดสอบย่อย มีบรรทัดเดียว ให้แสดงจำนวนจริง 1 ตัว ซึ่งแสดงค่า Average Waiting Time โดยมีความละเอียดของคำตอบเป็นทศนิยม M ตำแหน่ง

ข้อควรรู้ การแสดงทศนิยมแบบกำหนดจำนวนตำแหน่งได้ เช่น printf(“%.*lf\n”, 8, 5.34); จะแสดง 5.34000000 ออกมา

Example :

Sample Input	Sample Output
2 5 20 3 3 5 2 4 10 2 8 7 3 4 1 3 5 5 6 2 10 1 7	2.800 1.5000000

อธิบายตัวอย่างที่ 1

มีทั้งหมด 2 ชุดทดสอบย่อย

ชุดทดสอบย่อยแรก มีทั้งหมด 5 โปรเซส และให้ปริ้นคำตอบด้วยทศนิยม 3 ตำแหน่ง

โปรเซสที่ 3 เกิดที่เวลา t = 2; ได้เข้าทำงานในซีพียูที่เวลา t = 2; ทำงานเสร็จที่เวลา t = 2 + 4 = 6; (รอ = 0)

โปรเซสที่ 2 เกิดที่เวลา t = 3; ได้เข้าทำงานในซีพียูที่เวลา t = 6; ทำงานเสร็จที่เวลา t = 6 + 5 = 11; (รอ = 3)

โปรเซสที่ 5 เกิดที่เวลา t = 8; ได้เข้าทำงานในซีพียูที่เวลา t = 11; ทำงานเสร็จที่เวลา t = 11 + 7 = 18; (รอ = 3)

โปรเซสที่ 4 เกิดที่เวลา t = 10; ได้เข้าทำงานในซีพียูที่เวลา t = 18; ทำงานเสร็จที่เวลา t = 18 + 2 = 20; (รอ = 8)

โปรเซสที่ 1 เกิดที่เวลา t = 20; ได้เข้าทำงานในซีพียูที่เวลา t = 20; ทำงานเสร็จที่เวลา t = 20 + 3 = 23; (รอ = 0)

Average Waiting Time = (0 + 3 + 3 + 8 + 0) / 5 = 2.800

ชุดทดสอบย่อยที่สอง มีทั้งหมด 4 โปรเซส และให้ปริ้นคำตอบด้วยทศนิยม 7 ตำแหน่ง

โปรเซสที่ 1 เกิดที่เวลา t = 1; ได้เข้าทำงานในซีพียูที่เวลา t = 1; ทำงานเสร็จที่เวลา t = 1 + 3 = 4; (รอ = 0)

โปรเซสที่ 2 เกิดที่เวลา t = 5; ได้เข้าทำงานในซีพียูที่เวลา t = 5; ทำงานเสร็จที่เวลา t = 5 + 5 = 10; (รอ = 0)

โปรเซสที่ 3 เกิดที่เวลา t = 6; ได้เข้าทำงานในซีพียูที่เวลา t = 10; ทำงานเสร็จที่เวลา t = 10 + 2 = 12; (รอ = 4)

โปรเซสที่ 4 เกิดที่เวลา t = 10; ได้เข้าทำงานในซีพียูที่เวลา t = 12; ทำงานเสร็จที่เวลา t = 12 + 1 = 13; (รอ = 2)

Average Waiting Time = (0 + 0 + 4 + 2) / 4 = 1.5000000