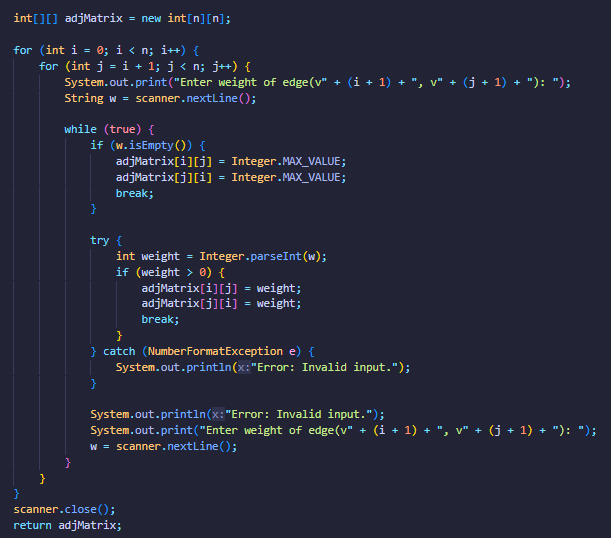
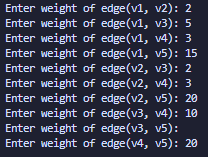
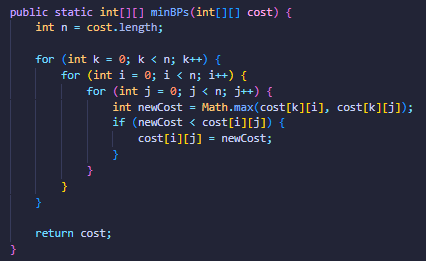


(1) ให้ผู้ใช้กรอกจำนวน vertices ที่ต้องการ ต้องกรอกเป็น Integer ถ้าหากผู้ใช้ไม่ได้กรอกเป็น Integer โปรแกรมจะแจ้งเตือน และให้ผู้ใช้ป้อนใหม่

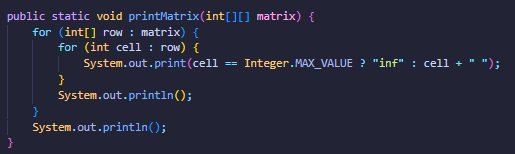




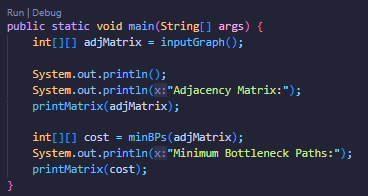
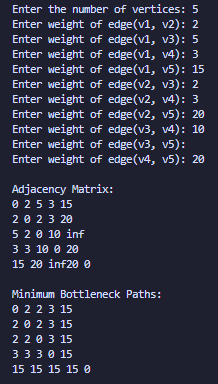
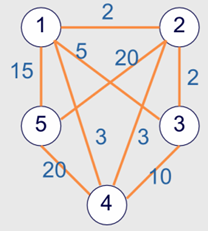
(2) สร้าง adjMatrix ชนาด n x n และให้ผู้ใช้กรอกค่า weight ของแต่ละเส้นเชื่อมในกราฟ โดยจะมีการตรวจสอบความถูกต้องในการป้อนด้วย คือ ผู้ใช้ต้องป้อนเฉพาะตัวเลข > 0 และสามารถป้อนข้อมูลว่างได้ เผื่อมีกรณีที่ไม่มีเส้นเชื่อม ถ้าหากกรอกผิด โปรแกรมจะเตือนและให้ป้อนใหม่ ซึ่งกระบวนการนี้จะใช้ for loop ในการรับค่า



(3) สร้าง matrix cost เพื่อเปรียบเทียบค่า weight ระหว่างจุด i และ j กับค่าที่สามารถทำให้เส้นเชื่อมที่ผ่านจุด k มีค่าน้ำหนักน้อยที่สุด หากพบว่าค่าใหม่น้อยกว่าค่าเดิม โปรแกรมจะอัพเดตค่าน้ำหนักใน matrix cost ด้วยค่าใหม่นั้น ๆ และส่ง matrix cost ที่อัพเดตแล้ว ไปใช้งานต่อไป



(4) แสดงผลลัพธ์ matrix โดยถ้าหากค่าใน matrix นั้นเป็น Integer.MAX\_VALUE (หมายถึงไม่มีเส้นเชื่อมระหว่างจุด) จะทำการใส่ inf ไว้ในค่านั้นแทน



(5) Main Program เรียกใช้ method ต่าง ๆ เพื่อรับค่า

และหลังจากนั้นจะทำการแสดงผลลัพธ์ในรูปแบบของ

Adjacency Matrix และ Minimum Bottleneck Paths