11. ตอบ. สามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจคกลับไปเป็นเวอร์ชั่นเก่าได้ นอกจากนั้นระบบ VCS ยังจะช่วยเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร ฯลฯ และยังช่วยให้สามารถกู้คืนไฟล์ที่ถูกลบหรือทำเสียโดยไม่ตั้งใจได้ หรือเรียกอีกอย่างว่า การ Backup

12. ตอบ. Distributed Version Control Systems (DVCSs) จะได้เปรียบกว่า Centralized Version Control Systems (CVCSs) ตรงที่การทำ DVCSs นั้นจะ copy ทั้งไฟล์ล่าสุดและ copy repository อีกด้วย  หมายความว่าถึงแม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository เหล่านี้ของ client ยังสามารถถูกก๊อปปี้กลับไปที่เซิร์ฟเวอร์เพื่อกูข้อมูลกลับคืนก็ได้ การ checkout แต่ละครั้งคือการทำสำรองข้อมูลทั้งหมดแบบเต็ม ๆ นั่นเอง

13. ตอบ. Centralized Version Control Systems (CVCSs) จะได้เปรียบกว่า Distributed Version Control Systems (DVCSs) ตรงที่ มีเซิร์ฟเวอร์กลางที่เก็บไฟล์ทั้งหมดไว้ในที่เดียวและผู้ใช้หลาย ๆ คนสามารถต่อเข้ามาเพื่อดึงไฟล์จากศูนย์กลางนี้ไปแก้ไขได้ และทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจคกำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง การจัดการแบบรวมศูนย์ในที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลใน client แต่ละเครื่อง

### 14. ตอบ. ใช้ merge source code แบบ GUI หรือ การแตก branch เป็นตัวช่วยในการแก้ปัญหา conflict จากการ merge ของโปรแกรม.

### 15. ตอบ. วิธีการลดปัญหา conflict ที่เกิดจากการ merge นั้นง่ายๆ คือ ถ้าเราทำงานกันหลายคนหรือเป็นทีมควรนั่งทำด้วยกันหรือมีการปรึกษาหารือกันตลอดเวลาเพื่อให้เข้าใจโค้ด หรือแก้ไขโค้ดที่ตรงกัน วิธีนี้เป็นวิธีที่ทำให้ลดการเกิด merge ได้เช่นกัน

### 16. ตอบ. **Git** คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ แต่ Github เป็นเว็บเซิฟเวอร์ที่ให้บริการในการฝากไฟล์ Git

### 17. ตอบ. branch เป็น feature ที่ช่วยให้นักพัฒนาสามารถที่จะทำงานได้สะดวกขึ้น เช่น ถ้าเราต้องการที่จะเพิ่มเติมหรือทดลองแก้ไขโค้ด โดยไม่ให้กระทบถึงโค้ดหลัก ก้เพียงแค่สร้าง branch ใหม่ขึ้นมา เมื่อแก้ไขหรือทำอะไรเสร็จแล้ว ก็ค่อยเซฟกลับมาที่ master เหมือนเดิม ดั้งนั้นการ branch ก้เปรียบเสมือนการสร้างโค้ดเสมือนตัวจริงขึ้นมาอีกโค้ดเพื่อทำการแก้ไขโค้ดนั้นๆ

### 18. ตอบ. Fast-forward คือถ้า Head ของเราอยู่ใน commits ที่ดึงมา, case นี้มักเกิดจากคำสั่ง "git pull" เพื่อดึง code จากต้นน้ำมา update code(ที่ไม่มีการเปลี่ยนแปลง) ของเรา, สิ่งที่เกิดขึ้นก็คือ git จะ update HEAD ของเราให้ตรงตาม HEAD ของ merged commit (โดยไม่มีการสร้าง commit object ใหม่ขึ้นมา)

### 19. ตอบ. คือรวมโค๊ดจาก remote มายัง local โดยที่เราไม่สามารถรู้ได้เลยว่าจะรวมโค๊ดอะไรบ้าง รู้แค่หลังจาก pull เสร็จแล้วนั่นเอง ซึ่งจริงๆแล้ว git pull มันก็คือการทำ git fetch และต่อด้วย git merge อัตโนมัตินั่นเอง

### 20. ตอบ.