# รายการอ้างอิง


ภาษาไทย

พูลพร แสงบางปลา. การบำรุงรักษาทวีผลสำหรับหัวหน้างาน. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), 2530.

_____. เอกสารเพิ่มประสิทธิภาพ. เอกสารประกอบการอบรม จุฬาลงกรณ์มหาวิทยาลัย , 2535.

ฟุกุนางะ อะจิโระ. เทคนิคการบำรุงรักษาเครื่องจักรกลในโรงงาน. แปลโดย ดร.ปริทรรศน์ พันธุบรรยงก์. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2530.

สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น). TOTAL PRODUCTIVE MAINTENANCE-TPM. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), 2534.

____. การบำรุงรักษาทวีผลที่ทุกคนมีส่วนร่วม เล่ม 1 แนวความคิดเกี่ยวกับการบำรุงรักษาเครื่องจักร. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น) , 2535.

____. การบำรุงรักษาทวีผลที่ทุกคนมีส่วนร่วม เล่ม 2 การคำนวนทางเศรษฐศาสตร์ของการบำรุงรักษาเครื่องจักร. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), 2535.

____. การบำรุงรักษาทวีผลที่ทุกคนมีส่วนร่วม เล่ม 3 การบำรุงรักษาเครื่องจักรเชิงปฏิบัติ. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น). 2535.

สมชาย พวงเพิกศึก. การบริหารงานซ่อมบำรุงสมัยใหม่. เอกสารประกอบการสัมนา สภาองค์การนายจ้างแห่งประเทศไทย, 2537.

อลงกฎ ชุตินันท์. การบำรุงรักษาทวีผลสำหรับหัวหน้างาน. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), 2537.

ภาษาอังกฤษ

Jt. Black. The design of the factory with a future. Auburn:
McGraw - Hill, inc., 1991.

# ภาคผนวก ก

# คู่มือการใช้โปรแกรมระบบสั่งการอัตโนมัติการบำรุงรักษาเครื่องจักรสำหรับงานหล่อลื่น

# ภาคผนวก ก.

## คู่มือการใช้โปรแกรมระบบสั่งการอัตโนมัติการบำรุงรักษาเครื่องจักรสำหรับงานหล่อลื่น

### 1. บทนำ

โปรแกรมนี้ สร้างขึ้นเพื่อใช้ในการพิมพ์ใบสั่งงานการหล่อลื่น (Lubricating Work Order) อย่างอัตโนมัติโดยไม่ต้องใช้คนในการค้นหาจุดหล่อลื่นที่ครบเวลาที่กำหนดเลยโดยโปรแกรมที่เขียนขึ้นมานี้ใช้ภาษา C++ ใช้ได้เฉพาะภาษาอังกฤษเท่านั้น ขนาดคอมพิวเตอร์ที่ใช้ควรเป็นรุ่น 386 หรือ 486 ส่วนเครื่องพิมพ์ควรมี Buffer อย่างน้อย 70 K.Byte เพื่อไม่ให้ข้อมูลจากคอมพิวเตอร์ที่ส่งเข้าเครื่องพิมพ์ต้องรอคอย (Queing) จะทำให้คอมพิวเตอร์ต้องหยุดทำงานชั่วคราว เป็นผลให้เวลาที่บันทึกลงในหน่วยความจำน้อยลงกว่าของจริง

### 2. การทำงานของโปรแกรม

โปรแกรมสามารถแบ่งออกเป็นหลัก ๆ ได้ 2 ส่วน คือ

#### 2.1 โปรแกรมการจัดการไฟล์หลัก (Master File)

เป็นโปรแกรมคอมพิวเตอร์ ที่ใช้ในการเก็บข้อมูลเริ่มต้นข้อมูลในที่นี้หมายถึงรายละเอียดต่าง ๆ เกี่ยวกับกิจกรรมต่าง ๆ ของจุดหล่อลื่นแต่ละจุดในเครื่องจักรแต่ละเครื่อง ในการศึกษานี่จะทำการใส่ข้อมูลดังกล่าวกับเครื่องจักร 3 เครื่อง คือ เครื่องเคลือบแล็คเกอร์, เครื่องขอยแผ่นเหล็ก และเครื่องทดสอบ (สมมติขึ้นเท่านั้น)

ไฟล์ดังกล่าว หรือไฟล์หลัก ใช้ชื่อว่า Poonsub.MAS ประกอบด้วยข้อมูลดังต่อไปนี้

1) Machine No. : /Machine Name : หมายถึงเครื่องจักรเบอร์อะไร,เครื่องจักรชื่ออะไร

2) Max.Point :/Check Poing : หมายถึงจุดหล่อลื่นที่มีในเครื่องจักรนั้น ๆ(ในโปรแกรมนี้ได้อนุญาตให้มีได้ไม่เกิน 25 จุด), จุดหล่อลื่นหรือจุดที่ต้องตรวจสอบเบอร์อะไร

3) Lubrication Code หมายถึง รหัสจุดหล่อลื่น

4) Part Name of M/C หมายถึง ชื่อชิ้นส่วนของเครื่องจักร

5) Specification of M/C หมายถึง ตำแหน่งหรือรายละเอียดของชิ้นส่วนในข้อ 4

6) Kind of Lubricant หมายถึงชนิดของสารหล่อลื่นคือ น้ำมัน(oil) หรือ จารบี(grease)

7) Grade of Lubricant หมายถึง สารหล่อลื่นที่ใช้เบอร์อะไร

8) Method Of Lubricant หมายถึงกิจกรรมการปฏิบัติสารหล่อลื่นคือ เปลี่ยนใหม่(replace) หรือเติมเพิ่ม (top-up)

9) Quantity Of Lubricant หมายถึง ปริมาณของสารหล่อลื่นที่ใช้ประมาณการ มีหน่วยเป็น litre สำหรับน้ำมัน และ gram สำหรับจารบี

10) Frequency Hour หมายถึงคาบเวลา หรือความถี่ที่ต้องปฏิบัติกิจกรรมการหล่อลื่น เช่น ทุก ๆ 500 ชม., ทุก ๆ 1,000 ชม., เป็นต้น

11) PM: Last Hour หมายถึงเพื่อแสดงให้รู้ว่ากิจกรรมที่ได้กระทำครั้งที่แล้วชั่วโมงที่เท่าไหร่

12) PM. Hour หมายถึง แสดงชั่วโมงการทำงานของเครื่องจักร ณ ปัจจุบัน

13) Next PM. Hour หมายถึง ชั่วโมงอะไรที่ต้องทำกิจกรรมในครั้งต่อไป ซึ่งเกิดจากการนำ PM. Last Hour+Frequency Hour

14) DD/MM/YY หมายถึง วัน-เดือน-ปี ที่ป้อนข้อมูลเข้าใน Cheer Point ครั้งล่าสุด

15) Done By Section หมายถึง แผนกอะไรเป็นผู้ทำกิจกรรมการหล่อลื่นนั้น ๆ

ข้อมูลที่เก็บในไฟล์ Poons.Mas ดูในภาคผนวก ฉ ส่วนไฟล์อื่น ๆ ที่ใช้ในการทำงานร่วมกับไฟล์หลักนี้ จัดเป็นกลุ่มหลัก ๆ ได้ 8 กลุ่มไฟล์ คือ

1) กลุ่ม ".C" เป็นไฟล์โปรแกรมที่เขียนขึ้นมาเองที่ Editor ในเทอร์โบซี ประกอบด้วย

-AUTO_WO1.C เป็นไฟล์โปรแกรมที่เขียนมาเพื่อใช้ ในการประมวลผลการทำงานต่าง ๆ ในระบบสั่งการอัตโนมัติ

-P_AUTO.C เป็นไฟล์โปรแกรมที่เขียนมาเพื่อใช้ในการจัดการเกี่ยวกับไฟล์หลัก(Master File) ที่ชื่อว่า Poons.Mas เพื่อเก็บข้อมูลมาตรฐานและรายละเอียดต่าง ๆ

2) กลุ่ม ".OBJ" เป็นไฟล์ที่เกิดจากการ Compile ไฟล์โปรแกรมที่เขียนด้วยเทอร์โบซี (ไฟล์ ชนิด C) เป็นไฟล์ชนิด .OBJ ประกอบด้วย P_AUTO.OBJ และ AUTO_WO1.OBJ

3) กลุ่ม ".EXE" เป็นไฟล์โปรแกรมที่ถูกสร้างขึ้นหลังจากการลิงส์ (Link) ระหว่างไฟล์ชนิด .OBJ เข้ากับไลบรารีมาตรฐานของเทอร์โบซีประกอบด้วย AUTO_WO1.EXE, P_AUTO.EXE

4) กลุ่ม ".H" เป็นไฟล์ที่เก็บไลบรารีมาตรฐาน (Standard Library) ของเทอร์โบซี ไลบรารีมาตรฐานจะถูกอ่านเข้ามารวมในโปรแกรมขณะที่ทำการคอมไฟล์ ประกอบด้วย Poons.H

5) กลุ่ม ".REP" เป็นไฟล์ที่ใช้ในการพิมพ์รายงานประจำเดือน คือ Month.rep

6) กลุ่ม ".CHR" เป็นไฟล์มาตรฐานในเทอร์โบซี ใช้ในการกำหนดรูปแบบตัวอักษร (Font)ประกอบด้วย GOTH.CHR, LITT.CHR, SANS.CHR, TRIP.CHR

7) กลุ่ม ".BGI" เป็นไฟล์มาตรฐานในเทอร์โบซีใช้ในการกำหนดจอภาพ

8) กลุ่ม ".MAS" เป็นไฟล์หลักที่เก็บข้อมูลต่าง ๆ ในการจัดการการหล่อลื่น ประกอบด้วย Poons.Mas และPoons_R.Mas, สะสมสำหรับ Poons_R.Mas เป็นไฟล์ที่นำข้อมูลหลักจาก Poons.Mas แต่ในPoons_R.Mas จะมีการเก็บจำนวนชั่วโมงการทำงานของเครื่องจักรแต่ละเครื่องไว้

ลำดับต่อไป จะเป็นการใช้โปรแกรมที่เขียนขึ้นในรูปของเมนู (Menu) ในการจัดการกับไฟล์หลัก ก่อนอื่นเข้าใน Subdirectory ที่มี Turbo c++ หรืออาจจะ Run ใน Subdirectory ที่สร้างขึ้นมาเอง แล้วใช้ .EXE ไฟล์ในการทำงาน สมมุติเข้าใน P_Auto.EXE จอภาพจะแสดงเป็น Menu ตามหน้าจอที่ 1

2.1.1 การสร้างไฟล์ข้อมูลหลักใหม่(New Created Data Base)

1) เลือก 'C' หน้าจอจะแสดงดังหน้าจอที่ 2

2) ใส่ชื่อเครื่องจักรเครื่องที่ 1 และ Enter

3) ใส่จำนวนจุดหล่อลื่นที่มี ที่ต้องการตรวจสอบในเครื่องจักรที่ 1 และ Enter

4) ใส่ชื่อเครื่องจักร และจำนวนจุดหล่อลื่นจนครบ 8 เครื่อง ถ้าไม่ต้องการครบ 8 เครื่อง เช่น ตามตัวอย่างต้องการ 3 เครื่อง วิธีให้เคาะ Enter ผ่านได้เลยจนได้ดังหน้าจอที่ 3

5) กรอกรายละเอียดต่าง ๆ ของเครื่องที่ 1 ตามหน้าจอที่ 3 ให้ครบ

6) ทำซ้ำข้อที่ 5 จนครบทุกเครื่อง

7) เคาะ Enter จนกลับไปที่หน้าจอที่ 1

8) ทำการบันทึกข้อมูลที่ได้ทำไว้ลงใน Hard disk โดย เลือก 'S' และตั้งชื่อไฟล์ที่จะบันทึก และ Enter ถ้าไม่พบปัญหาอะไรหน้าจอจะแสดงข้อความ "Sucess Save Data" ให้ Enter จนกลับมายังหน้าจอที่ 1

2.1.2 การเรียกแฟ้มข้อมูล(Load Data Base File)

1) เลือก 'L'

2) พิมพ์ชื่อไฟล์ที่ต้องการจะ Load ในหน่วยความจำ จากนั้น Enter จนกลับมายังหน้าจอที่ 1

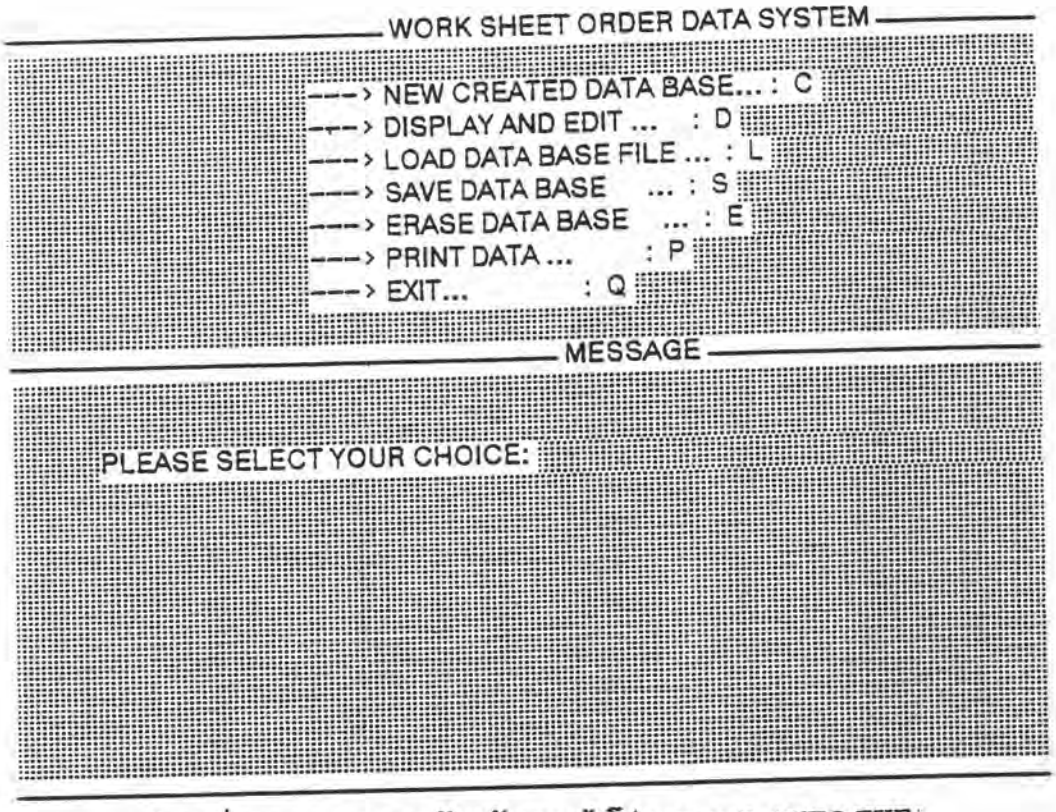
2.1.3 การแสดงผลและการแก้ไขข้อมูล(Display and Edit)

1) เลือก 'D' (ก่อนเลือก 'D' ต้อง Load แฟ้มข้อมูลในหัวข้อ 2.1.2 ก่อน)

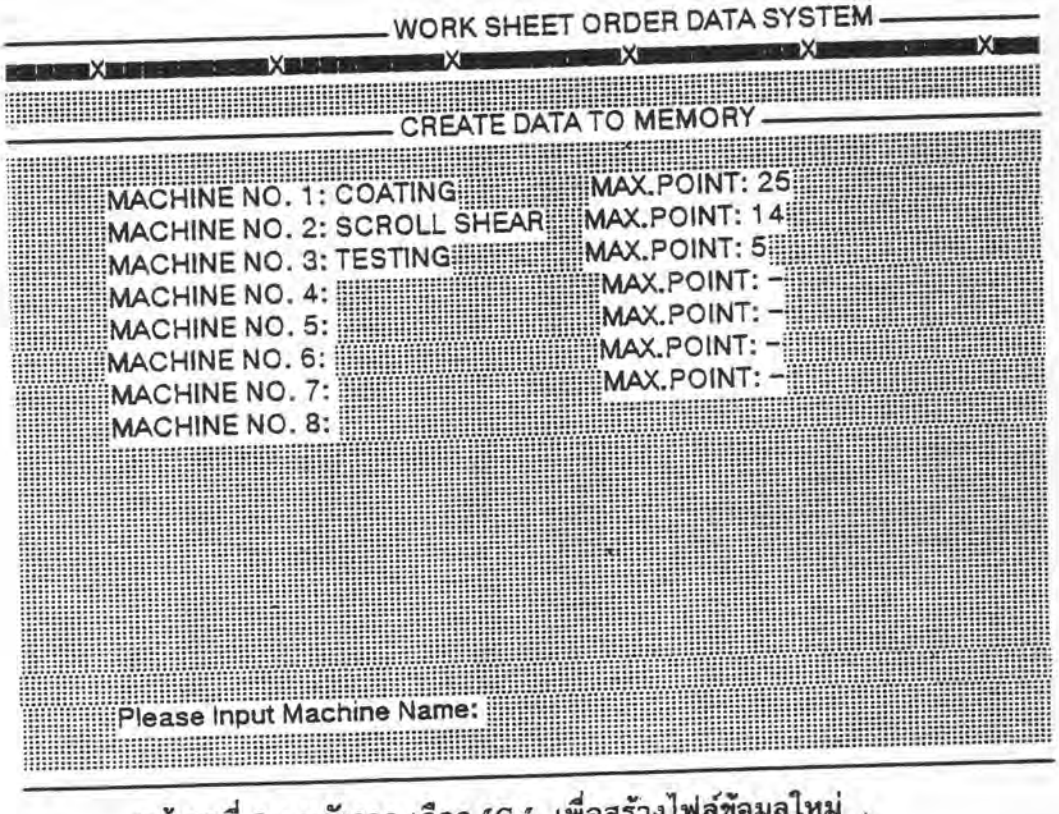
2) หน้าจอจะแสดงรายชื่อเครื่องจักรให้ดูพร้อมทั้งให้เลือกรายการ แสดงผล หรือแก้ไขข้อมูล(ดูหน้าจอที่ 4)

3) กรณีต้องการแสดงผลให้เลือก 't' เครื่องจะถามว่าจะขอดูเครื่องไหน(Please Input Machine No. )ในตัวอย่างเลือกเครื่องที่ 1 (ดูหน้าจอที่ 5) จากนั้นให้กดปุ่มใด ๆเพื่อดู Check Pointอื่น ๆ จนครบ จนกระทั่งเข้าสู่หน้าจอที่ 1

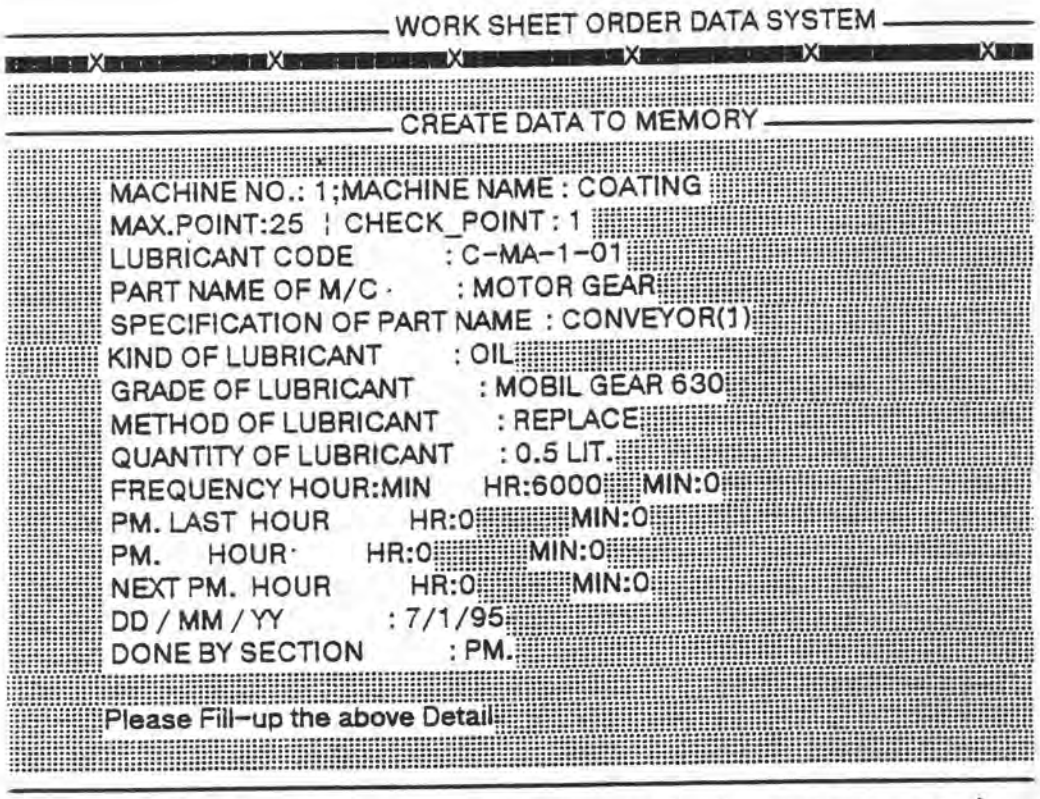
4) กรณีต้องการแก้ไขข้อมูลให้เลือด 'e' เครื่องจะถามว่าแก้ไขเครื่องไหน (Please



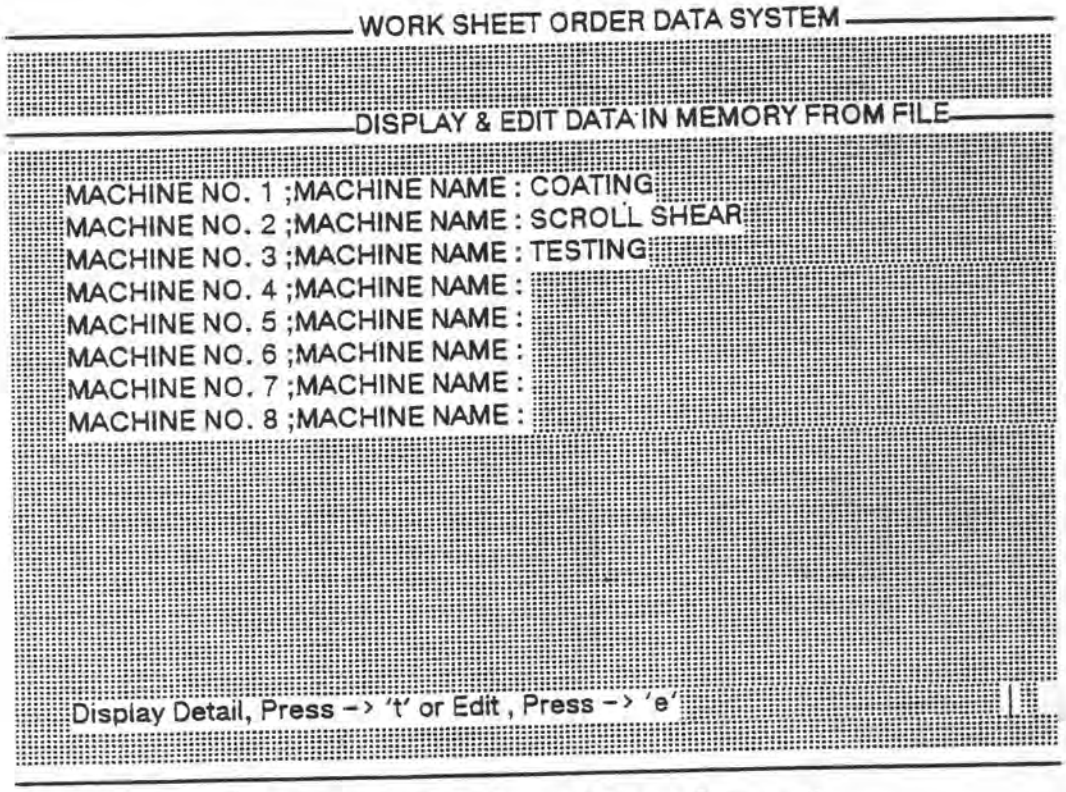

หน้าจอที่ 1 : แสดงเมนูหลักหลังจากเข้าโปรแกรม P_AUTO.EXE



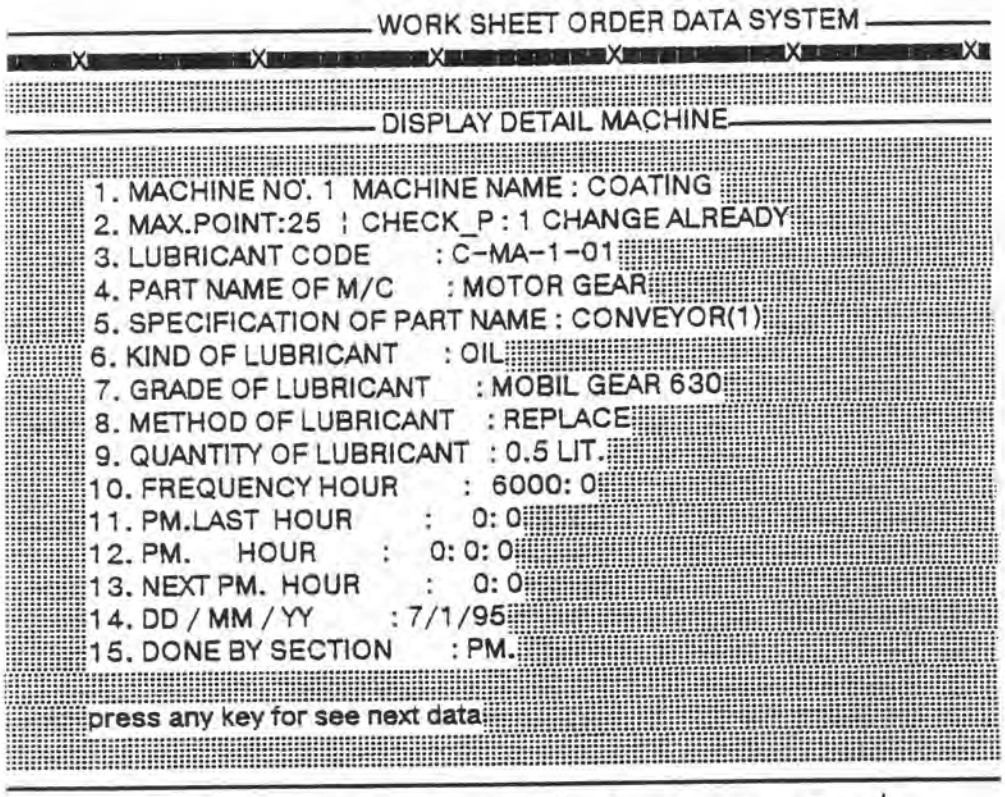

หน้าจอที่ 2 : หลังจาก เลือก "C" เพื่อสร้างไฟล์ข้อมูลใหม่



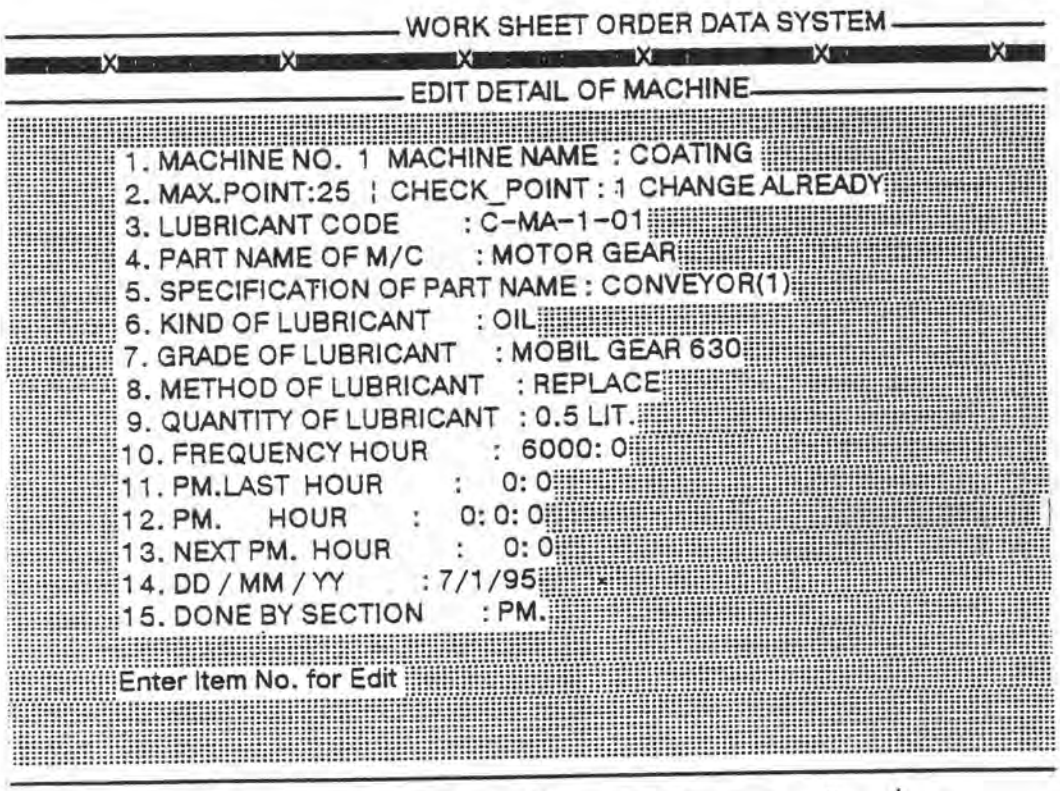

หน้าจอที่ 3 :เป็นตัวอย่างการกรอกข้อมูลรายละเอียดใน CHECK POINT ที่ 1 ของเครื่องเคลือบแลคเกอร์




หน้าจอที่ 4 : จอภาพแสดงหลังจากเลือก " D" ในเมนูหลัก




หน้าจอที่ 5 : แสดงข้อมูลต่าง ๆหลังจากเลือก ' t ' จากหน้าจอที่ 4




หน้าจอที่ 6 : แสดงข้อมูลต่าง ๆหลังจากเลือก ' E ' จากหน้าจอที่ 4

Enter Machine No. .) ในตัวอย่างเลือกเครื่องที่ 1 (ดูหน้าจอที่ 6) จากนั้นเครื่องจะถามว่าจะเข้าไปแก้ไขข้อมูลในหัวข้อไหน (Enter Item No. for Edit :) ให้ใส่ตัวเลขลงไป จากนั้น Enter จนกลับไปหน้าจอที่ 1 ใหม่

2.1.4 การบันทึกแฟ้มข้อมูลลงใน Hard Disk (Save Data Base)

ได้อธิบายไว้แล้วในหัวข้อ 2.1.1 ข้อ 8

2.1.5 การลบข้อมูล (Erase Data Base)

1) เลือก 'E'

2) หน้าจอจะแสดงตามหน้าจอที่ 7 พร้อมถามว่า ต้องการลบข้อมูลใช่หรือไม่ ถ้าต้องการให้เลือก 'E' ถ้าไม่ต้องการให้ Enter ไปเลย

3) กรณีเลือก 'E' และ Enter เครื่องจะถามว่าเครื่องจักรเบอร์อะไร(Please Input Machine No. :) ให้ใส่ตัวเลขพร้อม Enter จากนั้นเครื่องจะถามว่า จะลบที่ Check Point ไหนออก (Enter Check Point) ให้ใส่ตัวเลขที่ต้องการลบ Check Point นั้นๆออก พร้อม Enter

4) เครื่องจะถามว่าแน่ใจที่จะลบใช่หรือไม่? ถ้าต้องการลบให้เลือก 'y' ถ้าไม่ต้องการลบให้เลือก 'n' และ Enter

2.1.6 การพิมพ์ข้อมูล (Print Data)

1) เลือก 'P'

2) เครื่องจะถามว่า จะพิมพ์รายงาน จากเครื่องจักรเบอร์อะไร (Please Enter Machine No.:)

3) เลือกหมายเลขเครื่องจักร ทีละเครื่อง และกด Enter จะได้ข้อมูลครั้งละ 1 เครื่องจักร(อย่าลืมว่าจะต้องเปิดเครื่องพิมพ์ และใส่กระดาษไว้เรียบร้อยแล้ว)

2.1.6 การออกจากโปรแกรม (Exit)

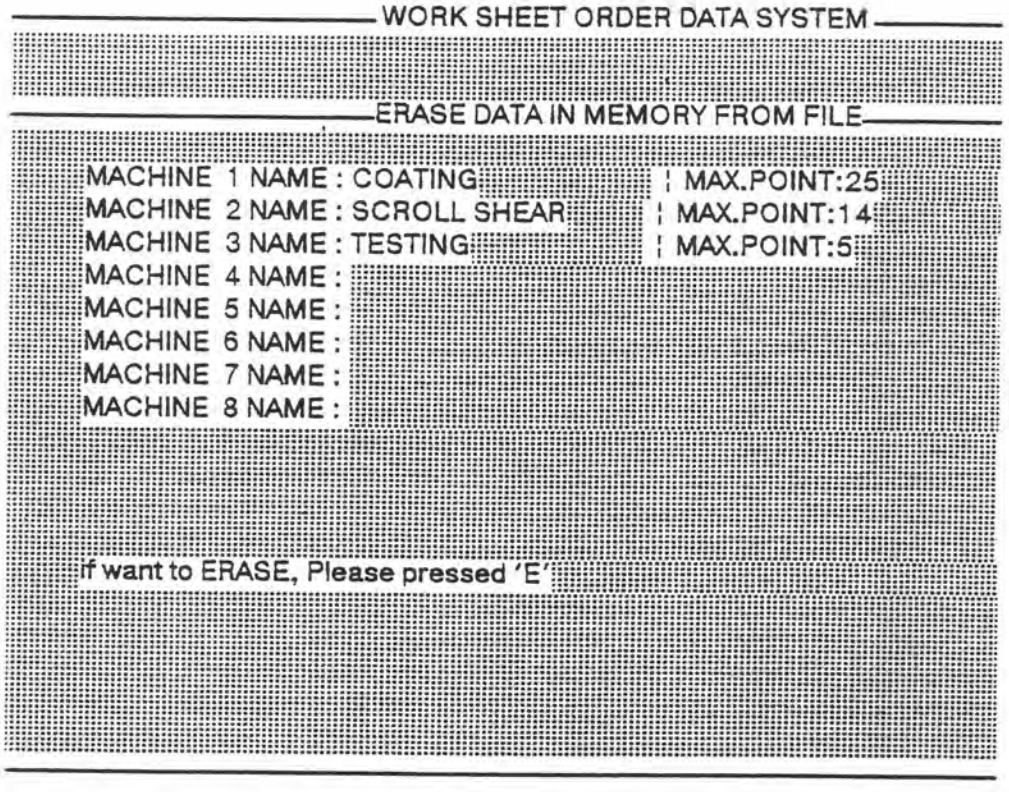
เลือก 'Q' เพื่อออกจาก Main Menu กลับไปยังตัวโปรแกรมกรณีเข้าทางTurbo C หรือกลับไปยัง Dos กรณี Run บน .EXE File

2.1.7 การเพิ่มข้อมูล

1) ให้เข้าไนโหมด Edit คือ เลือก 'D' จาก Main Menu จากนั้นเลือก 'E'

2) เลือกเบอร์เครื่องจักรและเลือก '2' เพื่อเข้าไปเพิ่ม MAX. Point เช่น เดิมมี 5 Check Point ต้องการเป็น 7 Check Point ให้ใส่ '7' และ Enter

3) จากนั้นให้กลับไปข้อที่ 1 และ เลือกเบอร์เครื่องจักรที่ 6 เป็นต้น จากนั้นใส่ข้อมูลลงในแต่ละรายการ ทีละรายการจนครบถ้วน




หน้าจอที่ 7 แสดงหน้าจอในโหมดลบ CHECK POINTหลังจากการเลือก'E'จากเมนูหลัก

## 2.2 โปรแกรมการสั่งงานอัตโนมัติ

เป็นโปรแกรมที่ใช้ในการประมวลผลต่าง ๆในไฟล์หลัก(POONS.MAS หรือPOONS_R.MAS) โดยมีข้อมูลเข้าจากแรงดันไฟฟ้าของเครื่องจักร ผ่านคอมพิวเตอร์ ที่มีโปรแกรม AUTO_WO1.EXE ที่เขียนขึ้นมาใช้ในการควบคุมการทำงานทั้งหมด จนกระทั่งพบเงื่อนไขต่าง ๆในโปรแกรมคอมพิวเตอร์จะสั่งให้เครื่องพิมพ์ พิมพ์ใบสั่งงาน ,ใบเตือน และรายงานประจำเดือน สำหรับงานที่เสร็จแล้วและงานที่ยังไม่ได้ทำ อย่างอัตโนมัติ

### 2.2.1 การเข้าในโปรแกรมการสั่งงานอัตโนมัติ

1) พิมพ์ AUTO_WO1.EXE กรณีอยู่ใน Dos หรือเลือกAUTO_WO1.C ใน Menu ของ TURBO C++ ถ้าเลือกผ่านใน Menu ของ TURBO C++ จะต้องเข้าโหมด Compile และ Run เสียก่อน

2) คอมพิวเตอร์จะถามว่าการทำงานนี้เป็นครั้งงานแรกใช่หรือไม่ หมายความว่า ถ้าตอบว่า 'y' : ไฟล์ที่ใช้งานนี้จะไม่มีจำนวนชั่วโมงทำงานของเครื่องจักรเลย กล่าวคือ เท่ากับศูนย์ทั้งหมด แต่ถ้าตอบ 'n' หมายถึงไฟล์ที่ต้องการใช้งานนี้จะมีจำนวนชั่วโมงการทำงานของเครื่องจักรที่สะสมอยู่เดิมแล้ว การ Run ใหม่ จะนำเอาเวลาเก่า บวกเวลาใหม่ที่จะ Run ต่อไป

3) หน้าจอ จะแสดงสภาวะการทำงาน (ON/OFF) ของเครื่องจักรแต่ละเครื่อง พร้อมทั้งแสดงวัน-เดือน-ปี-เวลา ในระบบ Dos และเวลาที่สะสมในแต่ละเครื่อง

4) กรณีเมื่อ Check Point ใด ๆ ครบชั่วโมงตามที่ Set ไว้ เครื่องพิมพ์จะพิมพ์ใบสั่งงานต่าง ๆ ออก รายละเอียดของรายงานต่าง ๆนี้ได้กล่าวไว้แล้วในบทที่ 6 หัวข้อที่ 6.4

### 2.2.2 การเข้าไป Update ข้อมูล

การ Update ในที่นี้หมายถึงเมื่อ Check Point ใด ๆครบกำหนดเวลา คอมพิวเตอร์จะสั่งให้เครื่องพิมพ์ พิมพ์ใบสั่งงาน หลังจากที่ช่าง PM. ได้เติมหรือเปลี่ยนสารหล่อลื่นเป็นที่เรียบร้อยแล้ว พนักงานที่ป้อนข้อมูลจะต้อง ทำการปรับปรุงข้อมูลเพื่อให้คอมพิวเตอร์รับรู้ว่า Check Point นั้น ๆ ได้รับการบำรุงรักษาแล้ว หลังจากปรับปรุงข้อมูล คอมพิวเตอร์จะทำการคำนวนชั่วโมงในครั้งต่อไปที่จะต้องทำ(NEXT PM.) โดยนำเอาชั่วโมงที่ปัจจุบันขณะปรับปรุง บวกกับความถี่ที่กำหนดไว้ การ Update กระทำได้ดังนี้

1) พิมพ์ Pass word ผ่านทาง Key Board โดยพิมพ์คำว่า "Poonsub38" 1 หรือ 2 ครั้ง จนหน้าจอแสดงสภาวะแต่ละ Check Point ว่าได้ทำกิจกรรมหล่อลื่นแล้วหรือไม่ ถ้าไม่จะบอกจำนวนครั้งให้ด้วย

2) เครื่องจะถามว่าจะปรับปรุงที่เครื่องจักรเบอร์อะไรหลังจากเลือกเบอร์เครื่องจักรได้แล้ว เครื่องจะถามต่อว่า Check Point เบอร์อะไร หลังจากเลือกเสร็จแล้วเครื่องจะถามว่า แน่ใจว่าได้เปลี่ยน/เติม สารหล่อลื่นแล้ว ถ้าแน่ใจให้ตอบ 'y' ถ้าไม่ต้องการปรับปรุงข้อมูลให้ตอบ 'n' หรือ Enter

3) เครื่องจะบอกว่า ถ้าจะออกไปที่ Dos ให้กด 'x' แต่ถ้าต้องการกลับไปทำงานอย่างเดิมให้กด Enter

# ภาคผนวก ข

## โปรแกรมคอมพิวเตอร์การจัดการไฟล์หลัก

```
// *** P_AUTO.C FEB 95 *** //
#include "poons.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#define  foot_line 22
int k_no(unsigned int limited,int kpx, int kpy);
char *proj[] = {
                    ---> NEW CREATED DATA BASE... : C ,
                    ---> DISPLAY AND EDIT ...    : D ,
                    ---> LOAD DATA BASE FILE ... : L ,
                    ---> SAVE DATA BASE     ... : S ,
                    ---> ERASE DATA BASE    ... : E ,
                    ---> PRINT DATA ...         : P ,
                    ---> EXIT...                : Q ,
                    };
char  ch;
char string[mc];
char xstring[mc];
FILE *stream,*fr;
float temp;
struct detail{
            unsigned int port_no;
            // io port number
            unsigned int bit_p;
            // value of bit

            unsigned int act_cp;
            // tell how many check point is active
            unsigned int chang[max_p];
            // 0:changed already 1:not change
            /* ------------------------------------*/
            char N_mach[mc];
            // name of machine
            char Lub_code[max_p][mc];
            // LUBRICATION CODE
            char Part_name[max_p][mc];
            // PART NAME OF M/C
            char Spec[max_p][mc];
            // SPECIFICATION OF PART NAME
            char K_Lub[max_p][mc];
            // KIND OF LUBRICANT
            char G_Lub[max_p][mc];
            // GRADE OF LUBRICANT
            char M_Lub[max_p][mc];
            // METHOD OF LUBRICAT
            char Q_Lub[max_p][mc];
            // QUANTITY OF LUBRICATN
```

```
        unsigned int Freq[max_p];
        // FREQUENCY (HOUR)
        unsigned int P_Freq[max_p];
        // POINT OF FREQUENCY
        unsigned int PM_LAST[max_p];
        // PM.LAST | HR.NO | DD/MM/YY
        unsigned int P_PM_LAST[max_p];
        // POINT OF PM_LAST
        unsigned int PM_HR[max_p];
        // PM_HR
        unsigned int P_PM_HR[max_p];
        // POINT OF PM_HR
        unsigned int S_PM_HR[max_p];
        // SECOND OF PM_HR
        char PM_DMY[max_p][mc];
        unsigned int NEXTPM[max_p];
        // PM.NEXT HR.NO
        unsigned int P_NEXTPM[max_p];
        // POINT OF NEXTPM
        char Section[max_p][mc];
        // DONE BY (SECTION)

};
struct detail mach[max_m],*pm;

unsigned i // 0 : no load data 1 : load already


int write_to_file(void)
{
        int test;
        //clear_string();
        //if ((stream = fopen("GROUP_1.MAS", "wb")) == NULL)
        /* open file TEST.$$$ */
        //strcpy(string,"poons.mas");
        if ((stream = fopen(string,"wb")) == NULL)
        {
                fprintf(stderr, "Cannot open output file.\n");
                return 1;
        }
        test = fwrite(&mach,sizeof(mach),1,stream);
        printf("Sucess for save data\n");
        fclose(stream);
        getch();
        return 0;
}
int read_datafile(void)
{       int test;
                clear_data_mach();
                clear_string();
                gets(string);
                gotoxy(11,16);
```

```
                    if ((fr = fopen(string,"rb")) == NULL)
                    {
                            //gotoxy (11,16);
                            fprintf(stderr,"can't open file for read\n");
                            return 1;
                    }
                    else
                     printf("READ DATA FILE NAME : %s success",&string);
                    test = fread(&mach,sizeof(mach),1,fr);
                    fclose(fr);
                    getch();
                    pain_screen();
                    return 0;
}
enter_data()
{
        struct detail mach[max_m],*ptm;
        unsigned int test,i,x;
        unsigned int number,point;
        ptm  = &mach;

                    printf("Enter MACHINE NUMBER : ");
                    x = 1;
                    do
                    {       scanf("%d",&number);
                            if (number<max_m)
                                    x = 0;
                            else
                            {       x = 1;
      printf("re enter MACHINE NUMBER\n");
                            }
                    }while(x==1);
       printf("MACHINE NUMBR : %d  MACHINE NUMBER OK!!!\n",number);
                    printf("Enter point of check : ");
                    do{
                            scanf("%d",&point);
                            if (point<max_p)
                                    x = 0;
                            else
                            {       x = 1;
                                    printf("re enter POINT OF CHECK\n");
                            }
                    }while(x==1);

                    printf("Enter Name of machine\n");
                    gets(string);
                    strcpy(&mach[i].N_mach,string);
                    printf("&mach[i].N_mach = %s\n",mach[0].N_mach);
                    printf("Enter port_no (in HEX code) : ");
                    scanf("%X",&mach[i].port_no);
                    printf("Enter bit_p (in HEX code)   : ");
                    scanf("%X",&mach[i].bit_p);
                    printf("Enter Part name : ");
```

```
                gets(string);
                strcpy(&mach[i].Part_name[point],string);
                getch();
                return 1;
}
int clear_string()
{       int i;
                for (i=0;i<mc;i++)
                        string[i] = '\0';
                return 1;
}
clear_xstring()
{       int i;
                for (i=0;i<mc;i++)
                        xstring[i] = '\0';
                return 1;
}

clear_data_mach()
{       static int m,cp,col;
        static int ix,iy,iz;
                for (ix=0;ix<8;ix++){
                        for (iy=0;iy<mc;iy++)
                                mach[ix].N_mach[iy] = '\0';
                }

                for (m=0;m<max_m;m++){
                        mach[m].act_cp = 0;
                        for (cp=0;cp<max_p;cp++)
                        {       for (col=0;col<mc;col++)
                                {
        mach[m].Lub_code[cp][col]=mach[m].Part_name[cp][col]='\0';
        mach[m].Spec[cp][col]=mach[m].K_Lub[cp][col]='\0';
        mach[m].G_Lub[cp][col]=mach[m].G_Lub[cp][col]='\0';
                                mach[m].Q_Lub[cp][col] = '\0'; // new;
                                }
                        }
                }
                for (m=0;m<max_m;m++)
                        for (cp=0;cp<max_p;cp++){
                mach[m].Freq[cp] = mach[m].P_Freq[cp] = 0;
                mach[m].PM_LAST[cp] = mach[m].P_PM_LAST[cp] = 0;
                mach[m].PM_HR[cp]   = mach[m].P_PM_HR[cp]   = 0;
                mach[m].S_PM_HR[cp] = 0;
                mach[m].NEXTPM[cp]  = mach[m].P_NEXTPM[cp]  = 0;

                                mach[m].chang[cp]   = 0;
                        }

                return 0;
}
int delete_chkp(int number,int n) //clear_data_mach()
{       static int m,cp,col;
```

```
        static int ix,iy,iz;
                m = number;
                cp = n;

                for (col=0;col<mc;col++)
                {
        mach[m].Lub_code[cp][col]=mach[m].Part_name[cp][col]='\0';
        mach[m].Spec[cp][col]=mach[m].K_Lub[cp][col]='\0';
        mach[m].G_Lub[cp][col]=mach[m].G_Lub[cp][col]='\0';
        mach[m].Q_Lub[cp][col] = '\0'; // new;
        mach[m].M_Lub[cp][col] = '\0';
        mach[m].Section[cp][col] = '\0';
                }

                mach[m].Freq[cp] = mach[m].P_Freq[cp] = 0;
                mach[m].PM_LAST[cp] = mach[m].P_PM_LAST[cp] = 0;
                mach[m].PM_HR[cp]  = mach[m].P_PM_HR[cp]  = 0;
                mach[m].S_PM_HR[cp] = 0;
                mach[m].NEXTPM[cp]  = mach[m].P_NEXTPM[cp]  = 0;
                mach[m].chang[cp]  = 0;

                return 0;
}

void displ// x_post, y_post
char *menu[];
int  x, y, count;
{
        register int i;

                for (i=0; i<count; i++) {
                        gotoxy(x, y++);
                        printf(menu[i]);
                }
}

void draw_line(int startx, int endx, int starty)
{
        register int i;
                for (i=startx+1; i<endx; i++) {
                        gotoxy(i, starty);
                        putch(196);
                }
}
void draw_border(int startx, int starty, int endx, int endy)
{
        register int i;
                for (i=startx+1; i<endx; i++) {
                        gotoxy(i, starty);
                        putch(196);
                        gotoxy(i, endy);
                        putch(196);
                }
```

```
                for (i=starty+1; i<endy; i++) {
                        gotoxy(startx, i);
                        putch(179);
                        gotoxy(endx, i);
                        putch(179);
                }

                gotoxy(startx, endy);   putch(192);
                gotoxy(endx, starty);   putch(191);
                gotoxy(en putch(217);
                gotoxy(startx, starty); putch(218);
}
int part1()             // old is main
{       int i;
                clrscr();
        //enter_data();         // enter data to memory by key
        //put_data_mach();    // put data of machine to memory
        printf("sizeof (mach)= %u\n",sizeof(mach));
                getch();
                /* is ok ! */
                write_to_file();
                clear_data_mach();

        printf("mach[0].Lub_code[0]=%s\n",mach[0].Lub_code[0]);
        printf("mach[0].G_Lub[0] = %s\n", mach[0].G_Lub[0]);
        printf("PROCESS read data\n");
        getch();

                read_datafile();

                printf("mach[1].Lub_code[0]=%s\n",mach[1].Lub_code[0]);
                printf("mach[1].G_Lub[0] = %s\n", mach[1].G_Lub[0]);
                printf("DATA IN FREQ\n");
                for (i=0;i<max_p;i++)
                printf("mach[1].Freq[%d] = %d\n", i,mach[1].Freq[i]);
                getch();
        for (i=0;i<max_p;i++)
                printf("mach[1].Section[%d]=%s\n",i,mach[1].Section[i]);
                getch();
        for (i=0;i<max_p;i++)
                printf("mach[1].Lub_code[%d]=%s\n",i,mach[1].Lub_code[i]);
                getch();
        for (i=0;i<max_p;i++)
                printf("mach[0].chang[%d]=%d\n",i,mach[0].chang[i]);
                return 1;
}
clear_text_post(px,py,cnt)
{       int i;
                gotoxy (px,py);
                for (i=0;i<cnt;i++)
                        printf(" ");
                return 1;
}
```

```
// Read the current cursor position
void read_cursor_xy (x, y)
char  *x, *y;
{           union REGS r;

                    r.h.ah = 3;   // read cursor position
                    r.h.bh = 0;   // video page
                    int86(0x10, &r, &r);
                    *x = r.h.dl;
                    *y = r.h.dh;
}
int  erase_data()
{           int i;
                    clrscr();
                    pain_screen();
                    draw_border (1, 1, 79, 24);
                    gotoxy (26, 1);
                    highvideo();
                    cprintf(" WORK SHEET ORDER DATA SYSTEM ");
                    draw_border (4, 4, 79, 24);
                    gotoxy (26, 4);
                    cprintf("ERASE DATA IN MEMORY FROM FILE");
                    normvideo();
                    erase_act();
                    return 0;
}
int erase_act()
{
          int i,number,n,checkp;
          char ch,ich;
                    normvideo();
          for(i=0;i<8;i++){
                    gotoxy(11,6+i);
                    printf("MACHINE %2d NAME : %s\n",i+1,mach[i].N_mach);
                              if(mach[i].act_cp>0){
                                        gotoxy(45,6+i);
                              printf(" | MAX.POINT:%d",mach[i].act_cp);
                              }
                    }
          gotoxy(11, foot_line-4);  // 22 is row for foot_line  message
                    // foot_line = 22 of row
                    printf("if want to ERASE, Please pressed 'E'");
                    ch = getch();
                    ch = tolower(ch);
                    /*
                    if (ch=='e'){
                              //edit_data(number,n);
                              //Edit_data(number,n);
                    }*/

                    if (ch=='e'){       // DETAIL ROUTINE
                              for(;;){
                                        repair_line(11,foot_line-4);
```

```
                        gotoxy(11,foot_line-4);
                        printf("Please Input Machine No.: ");

                        if (scanf("%d",&number)==0){
                                scanf("%*s");
                                continue;
                        }
                        if (number>8)
                                continue;
                        else
                        if ((number<=8)&&(number>0))
                                break;
                }
                for(;;){
                        repair_line(11,foot_line-3);
                        gotoxy(11,foot_line-3);
                        printf("ENTER CHECKPOINT: ");

                        if (scanf("%d",&n)==0){
                                scanf("%*s");
                                continue;
                        }
                        if (n>max_p)
                                continue;
                        else
                        if ((n<max_p)&&(n>0))
                                break;
                }
                number = number-1;
                n = n-1;
                gotoxy(11,foot_line-2);
                printf("Are you sure to Erase ? <y/n> :");
                ch = getch();
                ch = tolower(ch);
                if(ch=='y'){
                        putch(ch);
                        delete_chkp(number, n);
                }
                //repair_line(11,foot_line);
        }

        getch();
        normvideo();
        return 0;

}
int display_data()
{       int i;

        clrscr();
        pain_screen();
        draw_border (1, 1, 79, 24);
        gotoxy (26, 1);
        highvideo();
        cprintf(" WORK SHEET ORDER DATA SYSTEM ");
```

```
                draw_border (4, 4, 79, 24);
                gotoxy (26, 4);
                cprintf("DISPLAY & EDIT DATA IN MEMORY FROM FILE");
                normvideo();
                display_act();
                return 0;
}
int display_act()
{
        int i,number,n,checkp;
        char ch,ich;
                normvideo();
  for(i=0;i<8;i++){
                        gotoxy(11,6+i);
    printf("MACHINE NO.%2d ;MACHINE NAME : %s\n",i+1,mach[i].N_mach);
        }
        gotoxy(11, foot_line);  // 22 is row for foot_line  message
        printf("Display Detail, Press -> 't' or Edit , Press -> 'e'");
                ch = getch();
                if ((ch=='e')||(ch=='E')){
                        repair_line(11, foot_line);
                        gotoxy(11,10+i);
                        printf("EDIT MODE");
                        for(;;){
                                gotoxy(11,11+i);
                                printf("Please Enter Machine No. ");
                                if (scanf("%d",&number)==0){
                                        scanf("%*s");
                                        continue;
                                }
                                if (number>8)
                                        continue;
                                else
                                if ((number<=8)&&(number>0))
                                        break;
                        }
                        number = number-1;
                        repair_line(11,foot_line);
                 for(;;){
                        gotoxy(11,12+i);
                        printf("Enter Check Point No. of This Machine");

                                if (scanf("%d",&n)==0){
                                        scanf("%*s");
                                        continue;
                                }
                                if (n>max_p)
                                        continue;
                                else
                                if ((n<max_p)&&(n>0))
                                        break;
                        }
                        n = n-1;
```

```
                        repair_line(11,foot_line);

                        //display_detail(number,checkp,ich);
                        //edit_data(number,n);
                        Edit_data(number,n);
                }
                else
                if ((ch=='t')||('T')) // DETAIL ROUTINE
                        repair_line(11,foot_line);
                        for(;;){
                                //repair_line(11, foot_line);
                                gotoxy(11,10+i);
                                printf("DETAIL MODE");
                                gotoxy(11,10+i+1);
                                printf("Please Input Machine No.: ");

                                if (scanf("%d",&number)==0){
                                        scanf("%*s");
                                        continue;
                                }
                                if (number>8)
                                        continue;
                                else
                                if ((number<=8)&&(number>0))
                                        break;
                        }
                        number = number-1;
                        repair_line(11,foot_line);
                        display_detail(number,checkp,ich);
                }
                //tool_mark();
                getch();
                normvideo();
                return 0;
}

int display_detail(number, checkp,ch)
int number, checkp;
char ch;
{       int i;

        clrscr();
        pain_screen();
        draw_border (1, 1, 79, 24);
        gotoxy (26, 1);
        highvideo();
        cprintf(" WORK SHEET ORDER DATA SYSTEM ");
        draw_border (4, 4, 79, 24);
        gotoxy (26, 4);
        cprintf(" DISPLAY DETAIL MACHINE");
        normvideo();
        gotoxy(11,foot_line);
        printf("press any key for see next data");
```

```
gotoxy(11,6);
printf(" MACHIN %2d  NAME       : %s ",
                              number+1, mach[number].N_mach);
gotoxy(11,7);
printf(" CHECK POINT          : ");//%d ",checkp+1);
i = 2;
if (ch=='s')
 {
 gotoxy(fir_c,6+i++);
 printf(" LUBRICANT CODE       : %s",
                              mach[number].Lub_code[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" PART NAME OF M/C     : %s",
                              mach[number].Part_name[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" SPECIFICATION OF PART NAME : %s",
                              mach[number].Spec[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" KIND OF LUBRICANT    : %s",
                              mach[number].K_Lub[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" GRADE OF LUBRICANT   : %s",
                              mach[number].G_Lub[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" METHOD OF LUBRICANT  : %s",
                              mach[number].M_Lub[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" QUANTITY OF LUBRICANT  : %s",
                              mach[number].Q_Lub[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" FREQUENCY HOUR       : %6d:%2d",
 mach[number].Freq[checkp],mach[number].P_Freq[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" PM. LAST  HOUR       : %6d:%2d",
 mach[number].PM_LAST[checkp],mach[number].P_PM_LAST[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" xxPM.     HOUR       : %6d:%2d:%2d",
 mach[number].PM_HR[0],mach[number].P_PM_HR[0],
 mach[number].S_PM_HR[0]);
 //mach[number].PM_HR[checkp],mach[number].P_PM_HR[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" NEXT PM.  HOUR       : %6d:%2d",
 mach[number].NEXTPM[checkp],mach[number].P_NEXTPM[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" DD / MM / YY         : %s",
                              mach[number].PM_DMY[checkp]);
 gotoxy(fir_c,6+i++);
 printf(" DONE BY SECTION      : %s",
                              mach[number].Section[checkp]);
}
else{
 for(checkp=0;checkp<mach[number].act_cp;checkp++){
            i = 0;
```

```
            gotoxy(fir_c,6+i++);
            printf(" 1. MACHINE NO.%2d  MACHINE NAME : %s ",
                                        number+1, mach[number].N_mach);
            gotoxy(fir_c,6+i++);
            printf(" 2. MAX.POINT:%2d  | CHECK_P : %d ",
                                        mach[number].act_cp,checkp+1);
if(mach[number].chang[checkp]==0)
            printf("CHANGE ALREADY");
else
            printf("NOT CHANGE %d",mach[number].chang[checkp]);
            gotoxy(fir_c,6+i++);
            printf(" 3. LUBRICANT CODE        : %s",
            mach[number].Lub_code[checkp]);
            gotoxy(fir_c,6+i++);
            printf(" 4. PART NAME OF M/C      : %s",
            mach[number].Part_name[checkp]);
            gotoxy(fir_c,6+i++);
            printf(" 5. SPECIFICATION OF PART NAME : %s",
            mach[number].Spec[checkp]);
            gotoxy(fir_c,6+i++);
            printf(" 6. KIND OF LUBRICANT     : %s",
            mach[number].K_Lub[checkp]);
            gotoxy(fir_c,6+i++);
            printf(" 7. GRADE OF LUBRICANT    : %s",
            mach[number].G_Lub[checkp]);
            gotoxy(fir_c,6+i++);
            printf(" 8. METHOD OF LUBRICANT   : %s",
                                        mach[number].M_Lub[checkp]);
             gotoxy(fir_c,6+i++);
             printf(" 9. QUANTITY OF LUBRICANT  : %s",
                                        mach[number].Q_Lub[checkp]);
             gotoxy(fir_c,6+i++);
             printf("10. FREQUENCY HOUR       : %6u:%2d",
             mach[number].Freq[checkp],mach[number].P_Freq[checkp]);
             gotoxy(fir_c,6+i++);
             printf("11. PM.LAST  HOUR        : %6d:%2d",
 mach[number].PM_LAST[checkp],mach[number].P_PM_LAST[checkp]);
             gotoxy(fir_c,6+i++);
             printf("12. PM.      HOUR        : %6d:%2d:%2d",
             mach[number].PM_HR[0],mach[number].P_PM_HR[0],
             mach[number].S_PM_HR[0]);
//mach[number].PM_HR[checkp],mach[number].P_PM_HR[checkp]);
             gotoxy(fir_c,6+i++);
             printf("13. NEXT PM.  HOUR       : %6d:%2d",
 mach[number].NEXTPM[checkp],mach[number].P_NEXTPM[checkp]);
             gotoxy(fir_c,6+i++);
             printf("14. DD / MM / YY         : %s",
                                        mach[number].PM_DMY[checkp]);
             gotoxy(fir_c,6+i++);
             printf("15. DONE BY SECTION      : %s",
                                        mach[number].Section[checkp]);
             tool_mark();
             getch();
```

```
                    //clr_spec_char(stx,endx,sty,endy)
                    clr_spec_char(38, 60, 6, 22);
            }
        }
                    getch();
                    return 0;
}

int edit_da// number: number of machine
{       int e_ir;
        int x,y,min_x, ny,min_c,hr_c;
                    clrscr();
                    pain_screen();
                    draw_border (1, 1, 79, 24);
                    gotoxy (26, 1);
                    highvideo();
                    cprintf(" WORK SHEET ORDER DATA SYSTEM ");
                    draw_border (4, 4, 79, 24);
                    gotoxy (26, 4);
                    cprintf(" EDIT DATA IN MEMORY  ");
                    normvideo();
                    e_ir = 6;
                    for(;;){
                            gotoxy(11,e_ir);
                            printf("Please enter number of machin : ");

                                if (scanf("%d",&number)==0){
                                        scanf("%*s");
                                        continue;
                                }
                                if (number>8)
                                        continue;
                                else
                                if ((number<8)&&(number>0))
                                        break;
                    }
                            number = number-1;
                            repair_line(11,foot_line);
                    gotoxy(11,e_ir++); //9);
                    printf("machine name is %s",mach[number].N_mach);
                    gotoxy(11,e_ir++); //10);
                    printf("enter CHECK-POINT of Freq : ");
                    ny = 11;
                    ny = key_input(ny, &hr_c);
                    x = hr_c;
                    gotoxy(11,11);
                    printf("FREQUENCY HOUR:MIN     : ");
                    ny = 12;
                    ny = key_hr_no(ny, &hr_c);
                    mach[number].Freq[x-1] = hr_c;
                    ny = key_min_no(ny, &min_c);
                    mach[number].P_Freq[x-1] = min_c;
                    gotoxy(11,13);
```

```
                printf("HOUR:MIN = %5d:%2d", mach[number].Freq[x-1],
                                        mach[number].P_Freq[x-1]);
                return 0;
}

int Edit_data(number, checkp)
int number, checkp;
{       int i;
        int f_rw;
        clrscr();
        pain_screen();
        draw_border (1, 1, 79, 24);
        gotoxy (26, 1);
        highvideo();
        cprintf(" WORK SHEET ORDER DATA SYSTEM ");
        //draw_border (4, 4, 79, 24);
        draw_border (3, 3, 79, 24);
        gotoxy (26, 3);
        cprintf(" EDIT DETAIL OF MACHINE");
        normvideo();
        i = 2;
        f_rw = 5;
        if ((number<8)&&(checkp<=mach[number].act_cp))
        {
         //for(checkp=0;checkp<mach[number].act_cp;checkp++){
                i = 0;
                gotoxy(fir_c,f_rw+i++);
                printf(" 1. MACHINE NO. %2d  MACHINE NAME  : %s ",
                        number+1, mach[number].N_mach);
                gotoxy(fir_c,f_rw+i++);
                printf(" 2. MAX.POINT:%2d  | CHECK_POINT : %d ",
                        mach[number].act_cp,checkp+1);
                if(mach[number].chang[checkp]==0)
                  printf("CHANGE ALREADY");
                else
                  printf("NOT CHANGE %d",mach[number].chang[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 3. LUBRICANT CODE        : %s",
                                        mach[number].Lub_code[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 4. PART NAME OF M/C      : %s",
                                        mach[number].Part_name[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 5. SPECIFICATION OF PART NAME : %s",
                                        mach[number].Spec[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 6. KIND OF LUBRICANT     : %s",
                                        mach[number].K_Lub[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 7. GRADE OF LUBRICANT    : %s",
                                        mach[number].G_Lub[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 8. METHOD OF LUBRICANT   : %s",
```

```
                                        mach[number].M_Lub[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf(" 9. QUANTITY OF LUBRICANT  : %s",
                                        mach[number].Q_Lub[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf("10. FREQUENCY HOUR       : %6u:%2d",
                mach[number].Freq[checkp],mach[number].P_Freq[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf("11. PM.LAST  HOUR        : %6d:%2d",
        mach[number].PM_LAST[checkp],mach[number].P_PM_LAST[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf("12. PM.     HOUR         : %6d:%2d:%2d",
                mach[number].PM_HR[0],mach[number].P_PM_HR[0],
                mach[number].S_PM_HR[0]);
        //mach[number].PM_HR[checkp],mach[number].P_PM_HR[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf("13. NEXT PM.  HOUR       : %6d:%2d",
        mach[number].NEXTPM[checkp],mach[number].P_NEXTPM[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf("14. DD / MM / YY         : %s",
                                        mach[number].PM_DMY[checkp]);
                gotoxy(fir_c,f_rw+i++);
                printf("15. DONE BY SECTION      : %s",
                                        mach[number].Section[checkp]);
                tool_mark();
                //getch();
                Edit_keying(number, checkp,fir_c,f_rw+i);
  //clr_spec_char(38, 60, 6, 22); //clr_spec_char(stx,endx,sty,endy)
                // is function for clear part of screen
        }
                getch();
                return 0;
}
int  Edit_keying(int num, int chk,int px,int py)
{       int x,y;
        unsigned int z;
        char Ech[10];
                x = px;
                y = py;
                for(;;)
                {               gotoxy(x,y+1);
                                printf("Enter Item No. for Edit ");
                                if (scanf("%d",&z)==0){
                                        repair_line(x,y+1);
                                        scanf("%*s");
                                        continue;
                                }
                                if (z>15){
                                        repair_line(x,y+1);
                                        continue;
                                }
                                else
                                if ((z<=15)&&(z>0))
```

```
                        break;
}
//number = number-1;
//repair_line(11,foot_line);
gotoxy(x,y+2);
switch (z) {
case 1:
printf(" 1. MACHINE NO. %2d ; MACHINE NAME : ",num);
        k_string(num);
        strcpy(mach[num].N_mach, string);
        break;
case 2:
        printf(" 2. MAX.POINT: ");
        mach[num].act_cp = k_no(mc,x+30,y+2);
        //printf(" | CHECK_P : ");
        //mach[num].k_no(mc);
        break;
case 3:
        printf(" 3. LUBRICANT CODE      : ");
        k_string(num);
        strcpy(mach[num].Lub_code[chk], string);
        break;
case 4:
        printf(" 4. PART NAME OF M/C    : ");
        k_string(num);
        strcpy(mach[num].Part_name[chk], string);
        break;
case 5:
        printf(" 5. SPECIFICATION OF PART NAME : ");
        k_string(num);
        strcpy(mach[num].Spec[chk],string);
        break;
case 6:
        printf(" 6. KIND OF LUBRICANT   : ");
        k_string(num);
        strcpy(mach[num].K_Lub[chk], string);
        break;
case 7:
        printf(" 7. GRADE OF LUBRICANT  : ");
        k_string(num);
        strcpy(mach[num].G_Lub[chk], string);
        break;
case 8:
        printf(" 8. METHOD OF LUBRICANT : ");
        k_string(num);
        strcpy(mach[num].M_Lub[chk], string);
        break;
case 9:
        printf(" 9. QUANTITY OF LUBRICANT : ");
        k_string(num);
        strcpy(mach[num].Q_Lub[chk], string);
        break;
case 10:
```

```
                    printf("10. FREQUENCY HOUR      : ");
                    mach[num].Freq[chk] = k_no(65000,x+30,y+2);
                    gotoxy(x+40,y+2);
                    printf("MIN: ");
                    mach[num].P_Freq[chk] = k_no(60,x+45,y+2);
                    break;
                case 11:
                    printf("11. PM.LAST HOUR        : ");
                    mach[num].PM_LAST[chk] = k_no(65000,x+30,y+2);
                    gotoxy(x+40,y+2);
                    printf("MIN: ");
                    mach[num].P_PM_LAST[chk] = k_no(60,x+45, y+2);
                    break;
                case 12:
                    printf("12. PM.     HOUR        : ");
                    mach[num].PM_HR[0] = k_no(65000,x+30,y+2);
                    gotoxy(x+40, y+2);
                    printf("MIN: ");
                    mach[num].P_PM_HR[0] = k_no(60,x+45,y+2);
                    gotoxy(x+50, y+2);
                    printf("SEC: ");
                    mach[num].S_PM_HR[0] = k_no(60,x+55,y+2);
//mach[number].PM_HR[checkp],mach[number].P_PM_HR[checkp]);
                    break;
                case 13:
                    printf("13. NEXT PM. HOUR       : ");
                    mach[num].NEXTPM[chk] = k_no(65000,x+30,y+2);
                    gotoxy(x+40,y+2);
                    printf("MIN: ");
                    mach[num].P_NEXTPM[chk] = k_no(60, x+55, y+2);
                    break;
                case 14:
                    printf("14. DD / MM / YY        : ");
                    k_string(num);
                    strcpy(mach[num].PM_DMY[chk], string);
                    break;
                case 15:
                    printf("15. DONE BY SECTION     : ");
                    k_string(num);
                    strcpy(mach[num].Section[chk], string);
                    break;
                default:
                    gotoxy(x,y+2);
                    printf("out of case");
                    break;
                }
                return 1;
}
int k_string(int k_n)
{       int k_i;
        char k_ch[mc];
                clear_string();
                gets(string);
```

```
                gets(string);
                //strcpy(mach[k_n].N_mach,string);
                return 1;
}
int k_no(unsigned int limited,int kpx, int kpy)
{       unsigned long int no;
        int nx,i;

                for(;;){
                        gotoxy(kpx,kpy);

                        if (scanf("%ld",&no)==0){
                                scanf("%*s");
                                repair_line(kpx,kpy);
                                continue;
                        }
                        //scanf("%ld",&no);

                        if (no>=limited){
                                repair_line(kpx,kpy);
                                sound(3000);
                                delay(10);
                                nosound();
                                continue;
                        }
                        else
                        if ((no<=limited)&&(no>=0))
                                break;

                }
                return no;
}

void receive_key()
{       int i,chi;
                i = 0;
                clear_string();
                do{
                        chi = bioskey(0);
                        //chi = chi&0xff;

//printf("ASCII code is %xh, Scan code is %xh\n",chi&0xff,chi>>8);
                        //printf("          %c\n",chi);
                        //if (chi!=0x0d)
                        printf("%c",chi);
                        string[i++] = chi;

                        /*if ((chi&0xff) == 0x001b){
                                i = 0;
                                clear_string();
                                continue;
                        }*/
```

```
                    }while ((chi&0xff) != 0x0d);
                    //}while ((chi&0xff) != 0x001b);
                    string[i-1] = '\0';
}

int receive_knumber(col, row, p_no)
int *p_no;
int col, row;
{       int i,chi,cnt;

        unsigned char success_f,x,y;
                cnt = i = 0;
                clear_string();
                success_f = 1;  // 1 : means success
                do{

                        chi = bioskey(0);
                        cnt++;

 //printf("ASCII code is %xh, Scan code is %xh\n",chi&0xff,chi>>8);
                        //printf("       %c\n",chi);
                        //gotoxy(col++,row);
                        read_cursor_xy(&x,&y);
                        chi = chi&0xff;
                        if(chi==0x00)
                                continue;
                        if(chi!=0x0d)
                                printf("%c",chi);
                        //chi = chi&0xff;
                        if ((chi==0x0d)&&(cnt==1)){
                                cnt = 0;
                                success_f = 0;
                                gotoxy(x,y);
                                continue;
        }
                        if ((chi>=0x30)&&(chi<=0x39))
                                string[i++] = chi;
                        else{
                                i = 0;
                                if ((chi&0xff)!=0x0d)
                    success_f // 0 : means not success of proccess
                                continue;
                        }
                        /*if ((chi&0xff) == 0x001b){
                                i = 0;
                                clear_string();
                                continue;
                        }*/

                    }while ((chi&0xff) != 0x0d);// 0d is carriage return
                    //}while ((chi&0xff) != 0x001b);
                    string[i-1] = '\0';
                    *p_no = (atoi(string));
```

```
                return success_f;
}
int Getnum(p_no)
int *p_no;
{       int i,chi,cnt;

        unsigned char success_f,x,y;
                cnt = i = 0;
                clear_xstring();
                success_f = 1;  // 1 : means success
                do{

                        chi = bioskey(0);
                        cnt++;
//printf("ASCII code is %xh, Scan code is %xh\n",chi&0xff,chi>>8);
                        //printf("          %c\n",chi);
                        chi = chi&0xff;
                        if(chi!=0x0d)
                                printf("%c",chi);
                        if ((chi==0x0d)&&(cnt==1)){
                                cnt = 0;
                                success_f = 0;
                                gotoxy(x,y);
                                continue;
    }
                        if ((chi>=0x30)&&(chi<=0x39))
                                xstring[i++] = chi;
                        else{
                                i = 0;
                                if ((chi&0xff)!=0x0d)
                success_f // 0 : means not success of proccess
                                continue;
                        }

                }while ((chi&0xff) != 0x0d);// 0d is carriage return
                xstring[i-1] = '\0';
                *p_no = (atoi(xstring));
                return success_f;
}
int  create_data()
{       unsigned int hr_c, min_c, temp, temp_flag;
        unsigned char flag_flow;
        int i,nx,ny,ny_s,nx_s;
        int checkp,number;
        char x,y;   // x is column position : y is row position
        char old_x,old_y;
                clrscr();

                pain_screen();

                draw_border (1, 1, 79, 24);
                gotoxy (26, 1);
                highvideo();
```

```
cprintf(" WORK SHEET ORDER DATA SYSTEM ");
draw_border (4, 4, 79, 24);
gotoxy (26, 4);
cprintf(" CREATE DATA TO MEMORY ");
//gotoxy (11,foot_line);
//printf("Please Input Machine Name: ");
//----------------------
tool_mark();
//----------------------
normvideo();
clear_data_mach();
normvideo();
for(i=0;i<8;i++){
        repair_line(11,foot_line);
        gotoxy(11,foot_line);
        printf("Please Input Machine Name: ");
        gotoxy(fir_c,fir_r+i);
        printf("MACHINE NO. %d; MACHINE NAME : ",i+1);
        //gets(string);
        receive_key();
        strcpy(mach[i].N_mach,string);
        //read_cursor_xy(&x,&y);
        repair_line(11,foot_line);
        gotoxy(11,foot_line);
        printf("Please Input the Maximum Check Point");
        gotoxy(40,fir_r+i);
        printf("MAX.POINT: ");
        read_cursor_xy(&old_x,&old_y);
        //=============================================
        temp_flag = 5;
        if(string[0]!='\0'){
                x = y = 0;
                for(;;){
                        repair_line(old_x+1, old_y+1);
                        gotoxy(51,old_y+1);
                        temp_flag = Getnum(&temp);
                        if ((temp>max_p)||(temp<=0))
                                continue;
                        if (temp_flag==1){
                                mach[i].act_cp = temp;
                // put the value to ACTIVE CHECK.POINT
                                break;
                        }
                        else
                                continue;
                }
        }else
                printf("-");
}
        //=============================================
/***********************************************/

//gets(string);
```

```
clr_spec_char(fir_c, 70, fir_r, 23);
//gets(string);
// this point is deleting problem of clear buffer gets();
repair_line(11,foot_line);
gotoxy(11,foot_line);
printf("Please Fill-up the above Detail");
ny = fir_r;
for(number=0;number<8;number++){
        if (strncmp(mach[number].N_mach,string,3)!=0){
for(checkp=0;checkp<mach[number].act_cp;checkp++){
                        ny = fir_r;
                        //clear_string();
                        gotoxy(fir_c,ny++);

printf(" MACHINE NO.:%2d;MACHINE NAME : %s ",number+1,
mach[number].N_mach);
gotoxy(fir_c,ny++);
printf(" MAX.POINT:%2d | CHECK_POINT : %d ",
mach[number].act_cp,checkp+1);
clear_string();
gotoxy(fir_c,ny++);
printf(" LUBRICANT CODE          : ");
//gets(string);
receive_key();
strcpy(mach[number].Lub_code[checkp], string);
gotoxy(fir_c,ny++);
printf(" PART NAME OF M/C        : ");
clear_string();
receive_key();
strcpy(mach[number].Part_name[checkp], string);
gotoxy(fir_c,ny++);
printf(" SPECIFICATION OF PART NAME : ");
clear_string();
receive_key();
strcpy(mach[number].Spec[checkp], string);
gotoxy(fir_c,ny++);
printf(" KIND OF LUBRICANT       : ");
clear_string();
receive_key();
strcpy(mach[number].K_Lub[checkp], string);
gotoxy(fir_c,ny++);
printf(" GRADE OF LUBRICANT      : ");
clear_string();
receive_key();
strcpy(mach[number].G_Lub[checkp], string);
gotoxy(fir_c,ny++);
printf(" METHOD OF LUBRICANT     : ");
clear_string();
receive_key();
strcpy(mach[number].M_Lub[checkp], string);
gotoxy(fir_c,ny++);
printf(" QUANTITY OF LUBRICANT   : ");
                clear_string();
```

```
        receive_key();
    strcpy(mach[number].Q_Lub[checkp], string);
        ny_s = ny;

        gotoxy(fir_c,ny++);
        printf(" FREQUENCY HOUR:MIN        : ");
        ny = key_hr_no(ny, &hr_c);
        mach[number].Freq[checkp] = hr_c;
        ny = key_min_no(ny, &min_c);
        mach[number].P_Freq[checkp] = min_c;
        //mach[number].P_Freq[checkp] = 0;
        gotoxy(fir_c,ny++);
    printf(" PM. LAST  HOUR            : ");
        ny = key_hr_no(ny, &hr_c);
        mach[number].PM_LAST[checkp] = hr_c;
        ny = key_min_no(ny, &min_c);
        mach[number].P_PM_LAST[checkp] = min_c;

        gotoxy(fir_c,ny++);
    printf(" PM.      HOUR             : ");
        ny = key_hr_no(ny, &hr_c);
        mach[number].PM_HR[checkp] = hr_c;
        ny = key_min_no(ny, &min_c);
        mach[number].P_PM_HR[checkp] = min_c;

        gotoxy(fir_c,ny++);
    printf(" NEXT PM.  HOUR            : ");
        ny = key_hr_no(ny, &hr_c);
        mach[number].NEXTPM[checkp] = hr_c;
        ny = key_min_no(ny, &min_c);
        mach[number].P_NEXTPM[checkp] = min_c;

        gotoxy(fir_c,ny++);
    printf(" DD / MM / YY              : %s",
    mach[number].PM_DMY[checkp]);
        clear_string();
        receive_key();
    strcpy(mach[number].PM_DMY[checkp], string);

        gotoxy(fir_c,ny++);
    printf(" DONE BY SECTION           : %s",
        mach[number].Section[checkp]);
        receive_key();
    strcpy(mach[number].Section[checkp],string);
        //tool_mark();
        clr_spec_char(stx,endx,sty,endy)
                //gets(string);
                clear_string();
                clr_spec_char(38, 60, 6, 22);
        // start_x, end_x, start_y, end_y
        }
    }
}
```

```
                    // save_to_disk();
                    return 0;
};

int  key_input(ny, pt_m)    // this function is keying for
int ny;                 // only the hour value
int *pt_m;
{
        int ny_s;
        unsigned int hr_l, succ;
        unsigned int flag_f2;
                                hr_l = 0;
                                *pt_m = 0;
                                for (;;) {
    ny_s = ny;
                                        repair_line(fir_c+30,ny_s-1);
                                        //gotoxy(fir_c+36,ny_s-1);
                                        gotoxy(fir_c+27, ny_s-1);
                                        //printf("HR:");
                                succ = receive_knumber(fir_c+40,ny_s,&hr_l);
                                        //=============================================
                                        if (succ==1){
                                                if ((hr_l<0)||(hr_l>65000)){
                                                        flag_f2 = 0;
                                                repair_line(fir_c+36,ny_s-1);
                                                        continue;
                                                }else
                                                        flag_f2 = 1;
                                        }
                                        //=============================================
                                        else
                                                continue;
                                        if (flag_f2==1){
                                                ny = ny_s;
                                                break;
                                        }

                                }
                    if (flag_f2==1)
                                *pt_m = hr_l;
                    else
                                *pt_m = 0;
                    return ny;
}

int  key_hr_no(ny, pt_m)    // this function is keying for
int ny;                 // only the hour value
int *pt_m;
{
        int ny_s;
        unsigned int hr_l, succ;
        unsigned int flag_f2;
                                hr_l = 0;
```

```
                        *pt_m = 0;
                        for (;;) {
    ny_s = ny;
                                repair_line(fir_c+30,ny_s-1);
                                //gotoxy(fir_c+36,ny_s-1);
                                gotoxy(fir_c+27, ny_s-1);
                                printf("HR:");
                        succ = receive_knumber(fir_c+40,ny_s,&hr_l);
                                //=============================================
                                if (succ==1){
                                        if ((hr_l<0)||(hr_l>65000)){
                                                flag_f2 = 0;
                                        repair_line(fir_c+36,ny_s-1);
                                                continue;
                                        }else
                                                flag_f2 = 1;
                                }
                                //=============================================
                                else
                                        continue;
                                if (flag_f2==1){
                                        ny = ny_s;
                                        break;
                                }

                        }
                if (flag_f2==1)
                        *pt_m = hr_l;
                else
                        *pt_m = 0;
                return ny;
}

int key_min_no(ny,pt_m)  // this function is keying for
int ny;                  // only the minute  value
int *pt_m;
{
        int ny_s;
        unsigned int min_l, succ;
        unsigned int flag_f2;
                        min_l = 0;
                        *pt_m = 0;
                        for (;;) {
    ny_s = ny;
    repair_line(fir_c+40,ny_s-1);
                                gotoxy(fir_c+36,ny_s-1);
                                printf("MIN:");
                        succ = receive_knumber(fir_c+40,ny_s,&min_l);
                                //=============================================
                                if (succ==1){
                                        if ((min_l<0)||(min_l>59)){
                                                flag_f2 = 0;
                                        repair_line(fir_c+36,ny_s-1);
```

```
                                        continue;
                                }else
                                        flag_f2 = 1;
                        }
                        //=============================================
                        else
                                continue;
                        if (flag_f2==1){
                                ny = ny_s;
                                break;
                        }

                }
        if (flag_f2==1)
                *pt_m = min_l;
        else
                *pt_m = 0;
        return ny;
}

int keying_data()
{       int i,x,y;
        char ch;
                gotoxy (x,y);
                getch();
                return 0;
}

int tool_m// is tool box for marking screen monitor
{       int ix,iy,i;
                ix = 1;
                for (i=0;i<80;i++,ix++)
                {
                        gotoxy(ix,2);
                        //if ((i==10)||(i==20)||(i==30))
                        if ((i%10)==0)
                                printf("X");
                        else
                        printf("þ");
                }
                iy = 1;
                for (i=0;i<25;i++,iy++)
                {       gotoxy(2,iy);
                        if ((i%10)==0)
                                printf("X");
                        else
                        printf("þ");
                }
                return 1;
}
void screen_sys()
{
                clrscr();
```

```
                pain_screen();
                draw_border ( 1, 1, 79, 24);
                gotoxy(26,1);
                highvideo();
                cprintf(" WORK SHEET ORDER DATA SYSTEM ");
                //display_menu(proj, 22, 3, 6);
                // menu name , x_post, y_post, count
                display_menu(proj, 22, 3, 7);
                // menu name , x_post, y_post, count
                draw_line (1, 79, 11);
                gotoxy (35,11);
                printf(" MESSAGE ");
                normvideo();
}
int clear_MESS(py)
int py;
{       int x,y,i;
                //textcolor(3);
                //textbackground(3);
                for (i=py;i<24;i++){
                        gotoxy (2, i);
                        for (x=0;x<=75;x++)
                                putch(176);
                }
                return 0;
}
int  repair_line(col,row)
int row,col;
{  int x,y;
                textbackground(3);
                textcolor(1);
                gotoxy(row,col);
                for (x=col;x<79;x++) {
                        gotoxy(x,row);
                        cprintf("%c",176);
                }

}
int  pain_screen()
{       int i,j;
                textbackground(3);
                textcolor(1);
                j = 0;
                do{
                        for(i=0;i<80;i++){
                                cprintf("%c",176);
                                //cprintf(" ");
                        }
                        j++;
                }while(j<25);
                return 0;
}
```

```
int clr_spec_char(stx,endx,sty,endy)
int stx,endx,sty,endy;
{       int ix,iy,i;
                textcolor(1);
                textbackground(3);
                gotoxy(stx,sty);
                do{

                        for(i=stx;i<endx;i++)
                                cprintf("%c",176);
                        gotoxy(stx,sty++);

                }while(sty<endy);
                return 0;
}

void search_mach()
{       int i,found_f;
                found_f = 0;
                clear_string();
                gotoxy(11,16);
                printf("Name of machine : ");
                gets(string);
                gotoxy(11,17);
                for(i=0;i<8;i++){
                        if (strncmp (string, mach[i].N_mach, 4)==0){
                                printf("Found machine : ",mach[i].N_mach);
                                found_f = 1:
                        }
                        else
                                found_f = 0;
                }
                if (found_f == 0)
                        printf("Not found ! ");
                getch();
}
void save_data_mach()
{       int i,found_f;
                found_f = 0;
                clear_string();
                gotoxy(11,16);
                printf("File Name : ");
                gets(string);
                gotoxy(11,17);
                write_to_file();
                getch();
}
void print_data_machine()
{       unsigned int i,checkp,number;
                number = 0;
                //clear_string();
                for(;;){
                        gotoxy(11, 16);
```

```
                printf("Please enter Machine No.: ");
                scanf("%d",&number);
                if ((number>8)||(number<1)){
                        gotoxy(11,16);
                        printf("                              ");
                        continue;
                }
                else
                        break;
        }
        number = number-1;
        post_prn(5);
        fprintf(stdprn,"%c   DETAIL OF LUBRICATING ",14);
        nl_prn(1);
        post_prn(5);
fprintf(stdprn,"%c  MACHINE NAME:  %s ",14,mach[number].N_mach);
        nl_prn(1);
        for(checkp=0;checkp<mach[number].act_cp;checkp++){
         i = 0;
                nl_prn(0);
                first_col();
                post_prn(10);
fprintf(stdprn,"%c1. MACHINE NO.%2d|MACHINE NAME : %s ",15,
                number+1, mach[number].N_mach);
                nl_prn(0);
                first_col();
                post_prn(10);
        fprintf(stdprn,"%c2. MAX.POINT:%2d | CHECK_POINT : %d ",15,
                mach[number].act_cp,checkp+1);
                nl_prn(0);
                first_col();
                post_prn(10);
        fprintf(stdprn,"%c3. LUBRICANT CODE          : %s",15,
                                mach[number].Lub_code[checkp]);
                nl_prn(0);
                first_col();
                post_prn(10);
        fprintf(stdprn,"%c4. PART NAME OF M/C        : %s",15,
                                mach[number].Part_name[checkp]);
                nl_prn(0);
                first_col();
                post_prn(10);
        fprintf(stdprn,"%c5. SPECIFICATION OF PART NAME : %s",15,
                                mach[number].Spec[checkp]);
                nl_prn(0);
                first_col();
                post_prn(10);
        fprintf(stdprn,"%c6. KIND OF LUBRICANT       : %s",15,
                                mach[number].K_Lub[checkp]);
                nl_prn(0);
                first_col();
                post_prn(10);
        fprintf(stdprn,"%c7. GRADE OF LUBRICANT      : %s",15,
```

```
                                        mach[number].G_Lub[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(10);
                fprintf(stdprn,"%c8. METHOD OF LUBRICANT      : %s",15,
                                        mach[number].M_Lub[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(10);
                fprintf(stdprn,"%c9. QUANTITY OF LUBRICANT    : %s",15,
                                        mach[number].Q_Lub[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(9);
        fprintf(stdprn,"%c10. FREQUENCY HOUR          : %6u:%2d",15,
                mach[number].Freq[checkp],mach[number].P_Freq[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(9);
        fprintf(stdprn,"%c11. PM.LAST  HOUR           : %6d:%2d",15,
        mach[number].PM_LAST[checkp],mach[number].P_PM_LAST[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(9);
fprintf(stdprn,"%c12. PM.      HOUR           : %6d:%2d:%2d",15,
mach[number].PM_HR[0],mach[number].P_PM_HR[0],
mach[number].S_PM_HR[0]);
                        nl_prn(0);
                        first_col();
                        post_prn(9);
        fprintf(stdprn,"%c13. NEXT PM.  HOUR          : %6d:%2d",15,
        mach[number].NEXTPM[checkp],mach[number].P_NEXTPM[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(9);
                fprintf(stdprn,"%c14. DD / MM / YY            : %s",15,
                mach[number].PM_DMY[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(9);
                fprintf(stdprn,"%c15. DONE BY SECTION         : %s",15,
                                        mach[number].Section[checkp]);
                        nl_prn(0);
                        first_col();
                        post_prn(7);
                        fprintf(stdprn,"%c----------------------------
                        ----------",15);
                        nl_prn(0);
                }
                nl_prn(5);
}
// ********************************************
// ****** PRINTER TOOLS : move printing head ****
```

```
post_prn(int post)
{       int i,x;  // position of printer :int post = 0 -- 80 col.
                  fprintf(stdprn,"%c",13);
                  for (i=0;i<=post;i++)
                        fprintf(stdprn,"%c%s",18," ");
                  return 1;
}
// ***********************************************
// ******* PRINTER TOOLS : send newline  ********
nl_prn(int cnt_newline)// PRINTER TOOLS
{       int i;            // if cnt_newline = 0 is \n 1 count
                  for (i=0;i<=cnt_newline;i++)
                        fprintf(stdprn,"%c%s",18,"\n");
                  return 0;
}
first_col() // PRINTER TOOLS
{                 // This function is take the printer head at first post
                  fprintf(stdprn,"%c",13);
                  return 0;
}

out_to_print(number, cp)  // FUNCTION OF PRINTER PROCESS
int number, cp;
{ int i;
        char *msg[40] = {
LUBRICATING WORK ORDER,// 0
POOLSUB CAN CO,LTD.                    CC:ENGINEER DIVISION,//1
TO SECTION CHIEF.....  Date:.....□□Standard Time:.....,//2
Printed Date:.....  Completed Time:.....□□Hour No:.....,//3
Operating:          Results:,//4
Lub. Code  --->, //  5
Mic  Name  --->, //  6
Part Name  --->, //  7
Part Spec  --->, //  8

Lubricant  --->, //  9
Grade      --->, // 10
Method     --->, // 11
Quantity   --->, // 12

Frequency  --->, // 13
Last Hour No.--->, // 14
Pm.  Hour No.--->, // 15
SECTION CHIEF:.....  PREPARED  BY .....□□.....,//16
                PERFORMED  BY .....□□.....,//17
DIVISION MGR :.....  ACKNOWEDGED BY .....□□.....,// 18
[ ] RETURN TO ENGINEERING DIVISION,// 19
[ ] After this job finished, this data must be input to Computer in□□ order ,// 20
 to update this lub.code for the next due Entered by:.....□□.....,// 21
....., // 22
[ ] Complete, // 23
[ ] Comment , // 24
Remark :    // 25
```

```
};
        post_prn(15);
        fprintf(stdprn,"%c%s",14,msg[0]); //LUBRICANTING WORK ORDER
        nl_prn(3);
        fprintf(stdprn,"%c%c%s",27,69,msg[1]);//POONSUB CAN CO,LTD.
        nl_prn(0);
        fprintf(stdprn,"%c%c",27,70);
        nl_prn(1);
        //kbfprintf(stdprn,"%c",10); // carriage return

                fprintf(stdprn,"%c%s\n",15,msg[2]);
                nl_prn(0);
                fprintf(stdprn,"%c%s\n",15,msg[3]);
                nl_prn(1);
                fprintf(stdprn,"%c%c%s\n",27,69,msg[4]);
                fprintf(stdprn,"%c%c\n",27,70);
                fprintf(stdprn,"%c%s",15,msg[5]);// Lub. code
                // ----------LUBRICANT CODE ----------
                fprintf(stdprn,"%c%s",15,mach[number].Lub_code[cp]);
                // ------------------------------------

                post_prn(40);
                fprintf(stdprn,"%c%s",15,msg[23]);// [ ] Complete
                nl_prn(1);

                fprintf(stdprn,"%c%s",15,msg[6]);// Mic Name
                // --------- MACHINE NAME ----------
                fprintf(stdprn,"%c%s",15,mach[number].N_mach);
                // ------------------------------------
                post_prn(40);
                fprintf(stdprn,"%c%s",15,msg[22]);// .....
                nl_prn(1);

                fprintf(stdprn,"%c%s",15,msg[7]);// Part Name
                // ---------- PART NAME OF M/C ----------
                fprintf(stdprn,"%c%s",15,mach[number].Part_name[cp]);
                // ------------------------------------

                post_prn(40);
                fprintf(stdprn,"%c%s",15,msg[24]);// [ ] Comment
                nl_prn(1);

                fprintf(stdprn,"%c%s",15,msg[8]);// Part Spec
                // -------- SPECIFICATION PART NAME -------
                fprintf(stdprn,"%c%s",15,mach[number].Spec[cp]);
                // ------------------------------------
                post_prn(40);
                fprintf(stdprn,"%c%s",15,msg[22]);// .....
                nl_prn(1);

                fprintf(stdprn,"%c%s",15,msg[9]);// Lubricant
                // -------- KIND OF LUBRICANT ----------
```

```
fprintf(stdprn,"%c%s",15,mach[number].K_Lub[cp]);
// ------------------------------------

post_prn(40);
fprintf(stdprn,"%c%s",15,msg[22]);// .....
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[10]);
// ------- GRADE OF LUBRICANT ----------
fprintf(stdprn,"%c%s",15,mach[number].G_Lub[cp]);
// ------------------------------------


post_prn(40); // move the printing head to post of 40 unit
fprintf(stdprn,"%c%s",15,msg[22]);
nl_prn(1);   // new line

fprintf(stdprn,"%c%s",15,msg[11]);
// ------ METHOD OF LUBRICANT ---------
fprintf(stdprn,"%c%s",15,mach[number].M_Lub[cp]);
// ------------------------------------
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[12]);
// ------ QUANTITY OF LUBRICANT --------
fprintf(stdprn,"%c%s",15,mach[number].Q_Lub[cp]);
// ------------------------------------

post_prn(40);
fprintf(stdprn,"%c%s",15,msg[25]);
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[13]);
// --------- FREQUENCY HOUR ----------
fprintf(stdprn,"%c%d",15,mach[number].Freq[cp]);
// ------------------------------------

post_prn(40);
fprintf(stdprn,"%c%s",15,msg[22]);
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[14]);
// ---------- LAST HOUR NO. ----------
fprintf(stdprn,"%c%d",15,mach[number].PM_LAST[cp]);
// ------------------------------------

post_prn(40);
fprintf(stdprn,"%c%s",15,msg[22]);
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[15]);
// ---------- PM HOUR ----------
fprintf(stdprn,"%c%d",15,mach[number].PM_HR[cp]);
```

```
                    // ------------------------------------
                    nl_prn(2);


                    for (i=16;i<=21;i++)
                    {
                            fprintf(stdprn,"%c%s\n",15,msg[i]);
                            nl_prn(0);
                    }

                            //fprintf(stdprn,"%c%s\n",15,msg[6]);
                    //for (i=0;i<=8;i++)
                            //fprintf(stdprn,"%c%s\n",15,msg[i]);

                    return 1;

}

dot_to_post(int post,char *pt)   // PRINTER TOOLS
{       int i;
                    i = post;
                    post_prn(i);
                    fprintf(stdprn,"%c%s",15,pt);
                    //fprintf(stdprn,"%c%s",15,*pt);
                    return 0;
}

void main()
{ int i;
        char ch;
                    clrscr();
                    //part1();
                    screen_sys();
                    //tool_mark();
                    i = 0;
                    ch = '1';
                    while (ch!='q'){
                            gotoxy(11,14);
                            printf("PLEASE SELECT YOUR CHOICE: ");
                            ch = getch();

                            printf("%c",ch);
                            ch = tolower(ch);
            if( (ch>='a')&&(ch<= // skip over newline
                            switch (ch){
                                    case 'c':
                                            gotoxy (11,15);
                                            printf("C created process\n");
                                            create_data();
                                            screen_sys();
                                            break:
                                    case 'd':
                                            gotoxy (11,15);
```

```
                        printf("D display & edit \n");
                        display_data();
                        screen_sys();
                        normvideo();
                        break:
                case 'l':
                        gotoxy (11, 15);
                        printf("LOAD FILE NAME:");
                        read_datafile();
                        screen_sys();
                        break:
                case 's':
                        gotoxy (11, 15);
                        printf("S SAVE DATA \n");
                        save_data_mach();
                        //search_mach();
                        screen_sys();
                        break:
                case 'e':
                        gotoxy(11,15);
                        printf("E ERASER DATA \n");
                        erase_data();
                        screen_sys();
                case 'p':  break;

                        gotoxy (11, 15);
                        printf("P PRINT DATA \n");
                        print_data_machine();
                        //execl("G_SYS4.exe",NULL);
                        screen_sys();
                default:  break:

                        normvideo();
                        gotoxy (11,15);
                        printf("character out of case");
                        delay(500);
                        clear_MESS(13);
                        screen_sys();
                        }
        }
        gotoxy(11,14);
        printf("END OF PROCESS\n");
}
clrscr();

}
```

# ภาคผนวก ค

## โปรแกรมคอมพิวเตอร์การสั่งการอัตโนมัติ

```
// ========== GRAPHICS & AUTO_WO1.C ==========
// ==========    10 / 02 / 38    ==========
#define one 1
#define fin 30
#define ACTIVE 1
#define day_week "Tue"
#define day_month 1
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <time.h>
#include "poons.h"
#include <graphics.h>
#include <stdlib.h>
#include <ctype.h>
//#include "bios.h"
//#include "mem.h"

#define TIMER 0X1C
#define COM2  0X0B
#define COM1  0X0C
#define IRQ7  0x0F
#define CLKATTR 0X1E

char *proj[] = {
                --> NEW CREATED DATA BASE... : C,
                --> UPDATE DATA BASE ...    : U,
                --> READ DATA BASE FILE ... : R,
                --> SEEK DATA BASE   ...    : S,
                --> EXIT...                 : Q,
                };
struct detail{
        unsigned int port_no;
        // io port number
        unsigned int bit_p;
        // value of bit

        unsigned int act_cp;
        // tell how many check point is active
        unsigned int chang[max_p];
        // 0:changed already 1:not change
        /* ----------------------------------------*/
        char N_mach[mc];
        // name of machine
        char Lub_code[max_p][mc];
        // LUBRICATION CODE
        char Part_name[max_p][mc];
        // PART NAME OF M/C
        char Spec[max_p][mc];
        // SPECIFICATION PART NAME
```

```
        char K_Lub[max_p][mc];
        // KIND OF LUBRICANT
        char G_Lub[max_p][mc];
        // GRADE OF LUBRICANT
        char M_Lub[max_p][mc];
        // METHOD OF LUBRICANT
        char Q_Lub[max_p][mc];
        // QUANTITY OF LUBRICANT

        unsigned int Freq[max_p];
        // FREQUENCY (HOUR)
        unsigned int P_Freq[max_p];
        // POINT OF FREQUENCY
        unsigned int PM_LAST[max_p];
        // PM.LAST | HR.NO | DD/MM/YY
        unsigned int P_PM_LAST[max_p];
        // POINT OF PM_LAST
        unsigned int PM_HR[max_p];
        // PM_HR
        unsigned int P_PM_HR[max_p];
        // POINT OF PM_HR
        unsigned int S_PM_HR[max_p];
        // SECOND OF PM_HR

        char PM_DMY[max_p][mc];
        unsigned int NEXTPM[max_p];
        // PM.NEXT HR.NO
        unsigned int P_NEXTPM[max_p];
        // POINT OF NEXTPM
        char Section[max_p][mc];
        // DONE BY (SECTION)
};

struct detail mach[max_m],*pm;
char ch;
char string[mc];
char ch_string[80];
float temp;
int process(int number);
FILE *fr,*stream;
struct time t;
int bit_port[8];
unsigned int condition_f;
//---------- Graphic variable ----------
int maxx,maxy;
char txt[] = {"WORK SHEET ORDER SYSTEM"};
char *status[] = {"RUNNING","OFF"};
char *fkey[] = {"DATE:","TIME:","HOUR:","STATUS:"};
int time_o[7][5];
void Screen(int x1,int y1,int x2,int y2,char *str);
void Vbar(int x1,int y1,int x2,int y2);
void Vibar(int x1,int y1,int x2,int y2);
```

```
void Table(int c,int r);//,int stus);
void table(int x,int y);//,int stus);
void txt_Table(int c, int r, int stus);
void txt_table(int x, int y, int stus);
void SetTime(int c,int r, int h, int m, int s);
void OutTime(int c,int r);
void Out_STime(void);
void S_stus(int st);
void Out_hour(int hr,int min,int sec);
void Message(int n, int i);
int  Weekly_chek();
int  Weekly_print();
int  Monthly_process();
// ********* INTERUPT VARIABLE ********************
extern int CLOCKROW = 00, CLOCKCOL =  70 ;
unsigned char far *vid_mem, *StartVideoMemory ;
void interrupt (*OldClock)(void) ; /* Interrupt function pointer */
void interrupt (*OldCom2)(void) ; /* Interrupt function pointer */
void interrupt (*OldCom1)(void) ; /* Interrupt function pointer */
void interrupt Clock() ;
void interrupt Com2() ;
void interrupt Com1() ;
int clkcount = 0 ;
union REGS timer ;
unsigned char datacom2 = 0x00 ;
unsigned char d = 0, c2 = 'A' ;
unsigned char mask ;
unsigned int tick, Xck_;
unsigned int xc;
unsigned int train_st[4];//[0]:number,[1]:checkpoint
// ACTIVE:1 NON:0       //[2]:display_flag
// ***** pass word part *****
int chk_pass(void);
int pp = 0;
char Password[] =   {"poonsub38"};
char pass[9];
// ************************************
// ***** Warning printing part ********
char d_string[30];
unsigned int p_warn, p_tr;
// ***** Monthly Report **************
unsigned int p_Month;   // Monthly flag
// ************************************
int initial_sys()
{
                    return 1;
}
// ***********************************************
// *****FUNCTION OF PROCESS FOR CHECK MACHINE****
// ***********************************************
int G_process_real(n_number)
int n_nur   // number is number of machine
```

```
{
    static unsigned int xs,hour_sim, min_sim,sec_sim, i,ix;
    unsigned int sec_old, sec_now;
    unsigned int result, dummy,number,cnt, Hour_x, Min_x;
    char ch[] = "checking";
    //time_t tt;
    int s = 1;
    ix = result = 0;
    xs = n_number;
    //mov_freq_to_nextpm(0);
    //mov_freq_to_nextpm(1);
    //mov_freq_to_nextpm(2);
    //tool_mark();
    if (condition_f==0){
            mov_freq_to_nextpm(0);
            mov_freq_to_nextpm(1);
            mov_freq_to_nextpm(2);
    }
    hour_sim = 0;
    min_sim = 0;
    sec_sim = 0;
    sec_now = sec_old = 0;
    //p_warn = 1;
    //p_Month = 1;
    /*
    do{
            gettime(&t);
            Out_STime();
    }while(t.ti_sec!=59);
    delay(800);
    */
    putch('\07');
    do
    {
     //gettime(&t);
     Out_STime();
     if (tick==1)
     {      cnt = 1;
                    tick = 0;
     }
     else
                    cnt = 0;
     sec_sim = sec_sim+cnt;
     if (sec_sim>59){
                    sec_sim = 0;
                    min_sim = min_sim +1;
     }
     if (min_sim>59){
                    min_sim = 0;
                    hour_sim = hour_sim+1;
     }
     Out_hour(hour_sim,min_sim,sec_sim);
```

```
G_IODATA_input(0);
dummy = number;
for (number=0;number<3;number++)
{       if(bit_port[number]==ACTIVE)//MACHINE ACTIVE
                { // ------------------------------
                  for (i=0;i<mach[number].act_cp;i++)
                  {
  // freeze mach[number].PM_HR[i] = mach[number].PM_HR[i]+cnt;
  // new ----
  // mach[number].P_PM_HR[i] = mach[number].P_PM_HR[i]+cnt;
       mach[number].S_PM_HR[i] = mach[number].S_PM_HR[i]+cnt;
if (mach[number].S_PM_HR[i]>59)
  { mach[number].P_PM_HR[i] = mach[number].P_PM_HR[i] + 1;
       mach[number].S_PM_HR[i] = 0;
}
          if (mach[number].P_PM_HR[i]>59)
           {mach[number].PM_HR[i] = mach[number].PM_HR[i]+1;
           mach[number].P_PM_HR[i] = 0;
                        }
                        // new ----

         if((mach[number].PM_HR[0] == mach[number].NEXTPM[i])&&
          (mach[number].P_PM_HR[0]==mach[number].P_NEXTPM[i]))
                        //        &&(t.ti_sec==59))
                        // &&(mach[number].chang[i]==0))
                        {
mach[number].PM_LAST[i] = mach[number].NEXTPM[i];
mach[number].P_PM_LAST[i] = mach[number].P_NEXTPM[i];
mach[number].NEXTPM[i]  = mach[number].NEXTPM[i]+mach[number].Freq[i];
mach[number].P_NEXTPM[i] = mach[number].P_NEXTPM[i]+mach[number].
                    P_Freq[i];
      if(mach[number].P_NEXTPM[i]>=60)
       {mach[number].P_NEXTPM[i] = mach[number].P_NEXTPM[i]-60;
       mach[number].NEXTPM[i] = mach[number].NEXTPM[i]+1;
            }

            // ************ CHANGE FLAG ******************
             mach[number].chang[i] = mach[number].chang[i]+1;
            // ********************************************
                        putch(07);
                        SAVE_data_to_disk();
                        out_to_print(number,i);
                        train_st[0] = number;  // 0:number;
                        train_st[1] = i;       // 1:checkpoint;
          train_st[2] = 1;      // display flag
                        }

                  }
                } // ------------------------------
                else
                {  // XXX gotoxy(4,24);
                        // XXX printf("                ");
```

```
                        }
                }
                if (train_st[2]==1){
                        Message(train_st[0],train_st[1]);
                        train_st[2] = 0;
                }
                Weekly_check();
        }while(!chk_pass());
        s = 0;
        return 1;
}

int Weekly_check()
{ time_t t;
 struct date d;
                //putch('\07');
                time(&t);
                getdate(&d);
                strcpy(d_string,ctime(&t));
                //p_tr = strnicmp(d_string,"Tue",3);

                p_tr = strnicmp(d_string,day_week,3);

                //******* Weekly check part ********
                if ((p_tr == 0)&&(p_warn==1)){
                        putch('\07');
                        Weekly_print();
                        p_warn = 0;
                }
                if ((p_tr!=0)&&(p_warn==0))
                        p_warn = 1;
                //******* Monthly check part ********
                if ((d.da_day==day_month)&&(p_Month==1)){
                        putch('\07');
                        Monthly_print1();
                        Monthly_print2();
                        p_Month = 0;
                }
                if ((d.da_day!=day_month)&&(p_Month==0))
                        p_Month = 1;

                return 1;
}
int Weekly_print()
{       unsigned int Enumber,En;
                for (Enumber=0;Enumber<8;Enumber++){
                        for(En=0;En<mach[Enumber].act_cp;En++){
                                if(mach[Enumber].chang[En]>0){
                                 post_prn(0); // move printing head
                                                // to the first position
                                //fprintf(stdprn,"%cWARNING",15);
                                fprintf(stdprn,"%c** WARNING **",14);
```

```
                                        nl_prn(1);
                                        out_to_print(Enumber,En);
                                    }
                            }
                    }
                    return 1;
}
int Monthly_print1()
{       unsigned int Mnum, Mn, line_cnt;
                    line_cnt = 60;
                    post_prn(0);
                    nl_prn(1);
                    post_prn(0); // move printing head to first position
                    fprintf(stdprn,"%cMONTHLY REPORT (pending)\n",14);
                    nl_prn(0);
                    post_prn(0);
                    fprintf(stdprn,"%cDATE:%s",15,d_string);
                    nl_prn(0);
                    for(Mnum=0; Mnum<8; Mnum++){
                            for(Mn=0; Mn<mach[Mnum].act_cp; Mn++){
                                    if(mach[Mnum].chang[Mn]>0){
                                            post_prn(0);
    fprintf(stdprn,"%cMACHINE NAME:%14s | CHK_P:%2d | LUB. CODE %10s
    NOT CHANGE ",15,mach[Mnum].N_mach, Mn+1, mach[Mnum].Lub_code[Mn]);
                                            nl_prn(0);
                                            line_cnt = line_cnt - 1;
                                    }
                            }
                    }
                    nl_prn(line_cnt);
                    return 1;
}
int Monthly_print2()
{
        FILE *fppt;
        unsigned int Mnum, Mn, line_cnt;
                line_cnt = 60;
                post_prn(0);
                nl_prn(1);
                post_prn(0); // move printing head to the first position
                fprintf(stdprn,"%cMONTHLY REPORT (finished)\n",14);
                nl_prn(0);
                post_prn(0);
                fprintf(stdprn,"%cDATE:%s",15,d_string);
                nl_prn(0);
                if ((fppt = fopen("MONTH.REP","r"))==NULL){
                        printf("can't open file\n");
                        //break;
                }
                while(!feof(fppt)){
                        fgets(ch_string,80,fppt);
                        fprintf(stdprn,ch_string,14);
```

```
                        line_cnt = line_cnt - 1;
                        if(line_cnt<=0)
                                line_cnt = 60;
                };
                fclose(fppt);
                nl_prn(line_cnt);
                return 1;
}

int chk_pass(void)
{
unsigned char key;
int i;

        if(bioskey(1)){
                key = bioskey(0)&0xff;
                pass[pp] = key;
                pass[pp] = tolower(pass[pp]);
                if( pass[pp] == Password[pp] ){
                        pp++;
                }else{
                        pp=0;
                return(0);
                }
                if(pp == 9)  // OK return 1;
                        return(1);


        }
        return(0);
}
void Message(int Mn, int ki)
{
 int k;
 char ebuff[50];
 char sti[10];
                ki = ki+1;
                itoa(ki,sti,10);
                for(k=0;k<50;k++)
                        ebuff[k] = '\0';
                strcpy(ebuff,"MACHINE:");
                strcat(ebuff,mach[Mn].N_mach);
                strcat(ebuff," CHECKPOINT :");
                strcat(ebuff,sti);
                strcat(ebuff," Active");
                setcolor(RED);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
                outtextxy(25,420,ebuff);
                return;
}
int SAVE_data_to_disk()
{
```

```
        int test;
        //strcpy(string,"poons.mas");
        if ((stream = fopen("POONS_R.MAS","wb")) == NULL)
        {
                fprintf(stderr, "Cannot open output file.\n");
                return 1;
        }
        test = fwrite(&mach,sizeof(mach),1,stream);
        fclose(stream);
        return 0;
}
void initial_port()
{
                outportb(0x303, 0x92);
}
//*************************************************
// ****DATA ACQUISTION MODE: INPORTB FUNCTION*****
// ****  FOR GRAPHIC MODE **************************
//*************************************************
int G_IODATA_input(number)
int number;
{
        int i;
        char x,y;
                        x = inportb(0x300);
                        x = x&0xff;
                        y = 0xff-(x&0xff); // Real operation
                        //y = 0x02; // define for simulate of M/C #2
                        //y = 0x01;  // define for simulate of M/C #1
                        // bit D0
      // y & 0000 0001b  (0x01)
                        if ((y&0x1)==0x1){
                                //put_stat_on_block(10, 6, 1);
                                // (post_x:10, post_y:6, 1:on)
                                txt_Table(0, 0, 1);
                                bit_port[0] = 1;

                        }
                        else{
                                //put_stat_on_block(10, 6, 0);
                                txt_Table(0, 0, 0);
                                bit_port[0] = 0;

                        };
                        // ??????????????put_time_total(10,6+2,0);
                        //txt_Table(0, 0, 1);
                        SetTime(0,0,mach[0].PM_HR[0],mach[0].P_PM_HR[0],
                        mach[0].S_PM_HR[0]);
                        OutTime(0,0);

                        // bit D1
        // y & 0000 0010b  (0x02)
```

```
                    if ((y&0x2)==0x2){
                            //put_stat_on_block(28, 6, 1);
                            txt_Table(1, 0, 1);
                            bit_port[1] = 1;
                    }
                    else{
                            //put_stat_on_block(28, 6, 0);
                            txt_Table(1, 0, 0);
                            bit_port[1] = 0;
                    };
                    //put_time_total(28,6+2,1);
                    SetTime(1,0,mach[1].PM_HR[0],mach[1].P_PM_HR[0],
                    mach[1].S_PM_HR[0]);
                    OutTime(1,0);

                    // bit D2
// y & 0000 0100b  (0x04)
                    if ((y&0x4)==0x4){
                            //put_stat_on_block(46, 6, 1);
                            txt_Table(2,0,1);
                            bit_port[2] = 1;
                    }
                    else{
                            //put_stat_on_block(46, 6, 0);
                            txt_Table(2,0,0);
                            bit_port[2] = 0;
                    };
                    //put_time_total(46,6+2,2);
                    SetTime(2,0,mach[2].PM_HR[0],mach[2].P_PM_HR[0],
                    mach[2].S_PM_HR[0]);
                    OutTime(2,0);

                    // bit D3
// y & 0000 1000b  (0x08)
                    if ((y&0x8)==0x8){
                            //put_stat_on_block(64, 6, 1);
                            txt_Table(3,0,1);
                            bit_port[3] = 1;
                    }
                    else{
                            //put_stat_on_block(64, 6, 0);
                            txt_Table(3,0,0);
                            bit_port[3] = 0;
                    };

                    // bit D4
// y & 0001 0000b  (0x10)
                    if ((y&0x10)==0x10){
                            //put_stat_on_block(10,10, 1);
                            txt_Table(0,1,1);
                            bit_port[4] = 1;
                    }
```

```
                        else{
                                //put_stat_on_block(10,10, 0);
                                txt_Table(0,1,0);
                                bit_port[4] = 0;
                        };

                        // bit D5
    // y & 0010 0000b  (0x20)
                        if ((y&0x20)==0x20){
                                //put_stat_on_block(28,10, 1);
                                txt_Table(1,1,1);
                                bit_port[5] = 1;
                        }
                        else{
                                //put_stat_on_block(28,10, 0);
                                txt_Table(1,1,0);
                                bit_port[5] = 0;
                        };

                        // bit D6
    // y & 0100 0000b  (0x40)
                        if ((y&0x40)==0x40){
                                //put_stat_on_block(46, 10, 1);
                                txt_Table(2,1,1);
                                bit_port[6] = 1;
                        }
                        else{
                                //put_stat_on_block(46, 10, 0);
                                txt_Table(2,1,0);
                                bit_port[6] = 0;
                        };

                        // bit D7
                        // y & 1000 0000b (0x80)
                        if ((y&0x80)==0x80){
                                //put_stat_on_block(64, 10, 1);
                                txt_Table(3,1,1);
                                bit_port[7] = 1;
                        }
                        else{
                                //put_stat_on_block(64, 10, 0);
                                txt_Table(3,1,0);
                                bit_port[7] = 0;
                        };
                //}
                return 0;
}


show_timer()
{
                gotoxy (31,3);
```

```
                printf("%d:%d:%d  ", t.ti_hour, t.ti_min, t.ti_sec);
                return 0;
}
int display_hour(int c1,int c2 )

{
                gotoxy(45,3);
                highvideo();
                textbackground(3);
                cprintf("HOUR:");
                normvideo();
                gotoxy(50,3); printf("      ");
                gotoxy(50,3);
                printf("%5d:%2d",c1,c2);
                return 1;
}
// ***********************************************
// ***** MESSAGE DISPLAY WINDOWS **************
int mess_disp(int number, int point_c)
{               gotoxy (8, 22);
                highvideo();
                textbackground(4);
                cprintf ("REPORT:");
                normvideo();
        printf  ("NAME:%10s CHK_POINT:%2d --> PM_HOUR EQUAL NEXTPM\n",
                 mach[number].N_mach, point_c+1);
                gotoxy (8, 23);
                printf("                                                ");
                gotoxy (8,23);
                normvideo();
 cprintf ("     FREQ_HR=%4d:%2d | PM_HR=%4d:%2d | NEXT_PM=%4d:%2d",
           mach[number].Freq[point_c],mach[number].P_Freq[point_c],
           mach[number].PM_HR[0], mach[number].P_PM_HR[0],
           mach[number].NEXTPM[point_c],mach[number].P_NEXTPM[point_c]);
                normvideo();
                return 1;
}
// ***********************************************
// ******** CLEAR MESSAGE FUNCTION **************
int clear_mess_disp()
{   gotoxy (8, 22);
                printf("                                        ");
                gotoxy (8, 23);
                printf("                                        ");
                gotoxy (8, 24);
                return 1;
}
int mov_freq_to_nextpm(int number)
{       int i;

                for (i=0;i<mach[number].act_cp;i++){
                        mach[number].NEXTPM[i] = mach[number].Freq[i] ;
```

```
                        mach[number].P_NEXTPM[i] = mach[number].P_Frec
                }
        return 1;
}
void display_menu(menu, x, y, count)
char *menu[];
int  x, y, count;
{
        register int i;

                for (i=0; i<count; i++) {
                        gotoxy(x, y++);
                        printf(menu[i]);
                }
}

void draw_line(int startx, int endx, int starty)
{
        register int i;
                for (i=startx+1; i<endx; i++) {
                        gotoxy(i, starty);
                        putch(196);
                }
}
void draw_border(int startx, int starty, int endx, int endy)
{
        register int i;
                for (i=startx+1; i<endx; i++) {
                        gotoxy(i, starty);
                        putch(196);
                        gotoxy(i, endy);
                        putch(196);
                }
                for (i=starty+1; i<endy; i++) {
                        gotoxy(startx, i);
                        putch(179);
                        gotoxy(endx, i);
                        putch(179);
                }

                gotoxy(startx, endy);  putch(192);
                gotoxy(endx, starty);  putch(191);
                gotoxy(e putch(217);
                gotoxy(startx, starty); putch(218);
}

show_time(number,row,col,mss)// TIME & SHOW TOOLS
unsigned number,row,col;
char *mss;
{       static int i;
        char *temp;
                temp = mss;
```

```
                i = 0;
                strcat (mss,'\0');
                gotoxy (row, col);
                printf(":%5d HOUR",number);
                return 1;
}

// ***********************************************
// ****** PRINTER TOOLS : move printing head ****
post_prn(int post)
{       int i,x;  // position of printer :int post = 0 -- 80 col.
                fprintf(stdprn,"%c",13);
                for (i=0;i<=post;i++)
                        fprintf(stdprn,"%c%s",18," ");
                return 1;
}.
// ***********************************************
// ******* PRINTER TOOLS : send newline  *********
nl_prn(int cnt_newline)// PRINTER TOOLS
{       int i;              // if cnt_newline = 0 is \n 1 count
                for (i=0;i<=cnt_newline;i++)
                        fprintf(stdprn,"%c%s",18,"\n");
                return 0;
}
first_col( // PRINTER TOOLS
{                   // This function is to take the printer head at-
                    // the first post
                fprintf(stdprn,"%c",13);
                return 0;
}
out_to_print(number, cp)  // FUNCTION OF PRINTER PROCESS
int number, cp;
{ int i;
        char  *msg[40]  = {
LUBRICATING WORK ORDER,// 0
POONSUB CAN CO.,LTD.            CC:ENGINEERING DIVISION,//1
Printed Date: ,                 //2
Standard Time:....., //Completed Time:.....Hour No:
....." ,//3
Operating:                      Results:,//4
Lub. Code  —>, //  5
M/C  Name  —>, //  6
Part Name  —>, //  7
Part Spec.  —>, //  8

Lubricant  —>, //  9
Grade      —>, // 10
Method     —>, // 11
Quantity   —>, // 12

Frequency  —>, // 13
Last Hour No.—>, // 14
```

```
Pm. Hour No.--->, // 15
SECTION CHIEF:.....  PREPARED  BY .....,
//16
                    PERFORMED  BY .....,
//17
DIVISION MGR :.....  ACKNOWLEDGED BY .....,
// 18
[ ] RETURN TO ENGINEERING DIVISION,// 19
After this job finished, this data must be input to Computer in order ,
// 20
to update this lub.code for the next due. Entered by:.....,
// 21
....., // 22
[ ] Complete, // 23
[ ] Comment , // 24
Remark :    // 25

};
                post_prn(0);
                fprintf(stdprn,"%cCHECK POINT :%d",15,cp+1);
                nl_prn(0);
                post_prn(15);
                fprintf(stdprn,"%c%s",14,msg[0]); /LUBRICATING WORK ORDER
                nl_prn(3);

                fprintf(stdprn,"%c%c%s",27,69,msg[1]);//POONSUB CAN CO,LTD.
                nl_prn(0);
                fprintf(stdprn,"%c%c",27,70);
                nl_prn(1);
                //kbfprintf(stdprn,"%c",10); // carriage return

                fprintf(stdprn,"%c%s",15,msg[2]);
                fprintf(stdprn,"%c%s\n",15,d_string);
                nl_prn(0);
                fprintf(stdprn,"%c%s\n",15,msg[3]);
                //**********
                nl_prn(1);
                fprintf(stdprn,"%c%c%s\n",27,69,msg[4]);
                fprintf(stdprn,"%c%c\n",27,70);
                fprintf(stdprn,"%c%s",15,msg[5]);// Lub. code
                // ----------LUBRICANT CODE ----------
                fprintf(stdprn,"%c%s",15,mach[number].Lub_code[cp]);
                // ------------------------------

                post_prn(40);
                fprintf(stdprn,"%c%s",15,msg[23]);// [ ] Complete
                nl_prn(1);

                fprintf(stdprn,"%c%s",15,msg[6]);// Mic Name
                // ---------- MACHINE NAME ----------
                fprintf(stdprn,"%c%s",15,mach[number].N_mach);
                // ------------------------------
```

```
post_prn(40);
fprintf(stdprn,"%c%s",15,msg[22]);// .....
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[7]);// Part Name
// ———— PART NAME OF M/C ————
fprintf(stdprn,"%c%s",15,mach[number].Part_name[cp]);
// ————————————————

post_prn(40);
fprintf(stdprn,"%c%s",15,msg[24]);// [ ] Comment
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[8]);// Part Spec
// ——— SPECIFICATION PART NAME ———
fprintf(stdprn,"%c%s",15,mach[number].Spec[cp]);
// ————————————————
post_prn(40);
fprintf(stdprn,"%c%s",15,msg[22]);// .....
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[9]);// Lubricant
// ——— KIND OF LUBRICANT ————
fprintf(stdprn,"%c%s",15,mach[number].K_Lub[cp]);
// ————————————————

post_prn(40);
fprintf(stdprn,"%c%s",15,msg[22]);// .....
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[10]);
// ——— GRADE OF LUBRICANT ————
fprintf(stdprn,"%c%s",15,mach[number].G_Lub[cp]);
// ————————————————


post_prn(40); // move the printing head to post of 40 unit
fprintf(stdprn,"%c%s",15,msg[22]);
nl_prn(1);   // new line

fprintf(stdprn,"%c%s",15,msg[11]);
// ——— METHOD OF LUBRICANT ————
fprintf(stdprn,"%c%s",15,mach[number].M_Lub[cp]);
post_prn(40);
fprintf(stdprn,"%cDATE:.....",15);
// ————————————————
nl_prn(1);

fprintf(stdprn,"%c%s",15,msg[12]);
// ——— QUANTITY OF LUBRICANT ———

fprintf(stdprn,"%c%s",15,mach[number].Q_Lub[cp]);
```

```
post_prn(40);
                fprintf(stdprn,"Started time:.....",15);
                // ----------

                //post_prn(40);
                //fprintf(stdprn,"%c%s",15,msg[25]);
                nl_prn(1);

                fprintf(stdprn,"%c%s",15,msg[13]);
                // ---------- FREQUENCY HOUR ----------
                fprintf(stdprn,"%c%d:%d",15,mach[number].Freq[cp],
                 mach[number].P_Freq[cp]);
                // ----------

                post_prn(40);
                //fprintf(stdprn,"%c%s",15,msg[22]);
                fprintf(stdprn,"Finished time:.....",15);
                nl_prn(1);
                /*
                fprintf(stdprn,"%c%s",15,msg[14]);
                // ---------- LAST HOUR NO. ----------
                fprintf(stdprn,"%c%d:%d",15,mach[number].PM_LAST[cp],
                mach[number].P_PM_LAST[cp]);

                // ----------

                post_prn(40);
                fprintf(stdprn,"%c%s",15,msg[22]);

                nl_prn(1);
                */
                fprintf(stdprn,"%c%s",15,msg[15]);
                // ---------- PM HOUR ----------
                //fprintf(stdprn,"%c%d",15,mach[number].PM_HR[cp]);
                fprintf(stdprn,"%c%d:%d",15,mach[number].PM_HR[0],
                mach[number].P_PM_HR[0]);
                // ----------
                nl_prn(2);


                for (i=16;i<=21;i++)
                {
                        fprintf(stdprn,"%c%s\n",15,msg[i]);
                        nl_prn(0);
                }
                nl_prn(18);
                        //fprintf(stdprn,"%c%s\n",15,msg[6]);
                //for (i=0;i<=8;i++)
                        //fprintf(stdprn,"%c%s\n",15,msg[i]);

                return 1;
```

```
}

dot_to_post(int post,char *pt)   // PRINTER TOOLS
{       int i;
                i = post;
                post_prn(i);
                fprintf(stdprn,"%c%s",15,pt);
                return 0;
}
// ************************************************
// ******* CLEAR DATA STRUCTURE IN MEMORY *******
clear_data_mach()
{       static int m,cp,col,number,i;
        static int ix,iy,iz;
                for (ix=0;ix<8;ix++){
                        for (iy=0;iy<mc;iy++)
                                mach[ix].N_mach[iy] = '\0';
                }

                for (m=0;m<max_m;m++){
                        number = m;
                        mach[m].act_cp = 0;
                        for (cp=0;cp<max_p;cp++)
                        {       for (col=0;col<mc;col++)
                                {
            mach[m].Lub_code[cp][col]=mach[m].Part_name[cp][col]='\0';
            mach[m].Spec[cp][col]=mach[m].K_Lub[cp][col]='\0';
            mach[m].G_Lub[cp][col]=mach[m].G_Lub[cp][col]='\0';
                                }
                        }
                }
                for (m=0;m<max_m;m++)
                        for (cp=0;cp<max_p;cp++){
                        mach[m].Freq[cp] = mach[m].P_Freq[cp] = 0;
                        mach[m].PM_LAST[cp] = mach[m].P_PM_LAST[cp] = 0;
                        mach[m].PM_HR[cp]   = mach[m].P_PM_HR[cp]   = 0;
                        mach[m].NEXTPM[cp]  = mach[m].P_NEXTPM[cp]  = 0;
                        mach[m].chang[cp]   = 0;
                        }

                return 0;
}
void clear_string(void)
{       int i;
                for (i=0;i<mc;i++)
                        string[i] = '\0';
}
void clear_ch_string(void)
{       int i;
                for (i=0;i<80;i++)
                        ch_string[i] = '\0';
```

```
}

int read_datafile(void)
{       static int test;
                //clear_data_mach();
                clear_string();
                if(condition_f==0)
                        strcpy(string,"POONS.MAS");
                else
                        strcpy(string,"POONS_R.MAS");
                //if ((fr = fopen("POONS.MAS","rb")) == NULL)
                if((fr=fopen(string,"rb"))==NULL)
                {       fprintf(stderr,"can't open file for read\n");
                        return 0;
                }
                test = fread(&mach,sizeof(mach),1,fr);
                fclose(fr);
                return 1;
}
int read_poons_r(void)
{       static int test;
                if((fr=fopen("POONS_R.MAS","rb"))==NULL)
                {       fprintf(stderr,"can't open file for read\n");
                        return 0;
                }
                test = fread(&mach,sizeof(mach),1,fr);
                fclose(fr);
                return 1;
}
int screen_system()
{
        int co_col;
                co_col = 21;
                //      1, 1, 79, 25
                textcolor(4);       // part of put color on text
                textbackground(3);
                draw_border ( 1, 1, 79, 25); //(int startx, int starty,-
                                             //int endx, int endy)
                highvideo();
                gotoxy(26,1);
                cprintf(" MACHINE WORK ORDER SYSTEM ");
                normvideo();
                mach_block();
                draw_lin // startx, endx,
                gotoxy (35,co_col);
                printf(" MESSAGE ");
                w_on_block (1);
                getch();
            put_stat_on_block (10, 6, 1);  // x_post, y_post, 1:on 0:off
                return 1;
};
int time_date_screen()
```

```
{       static int px = 3;
        static int py = 10;
        struct date d;
                getdate(&d);
                gotoxy (8,3);
                highvideo();
                cprintf("DATE:");
                printf("%d:%d:%d" ,d.da_day, d.da_mon, d.da_year-1900);

                gotoxy (26,3);
                cprintf("TIME:");
                printf("xxxxx");

                gotoxy(45,3);
                cprintf("HOUR:");

                gotoxy(62,3);
                cprintf("STATUS:");
                printf("runing");
                return 1;
}
int  mach_block()
{
        int sx,sy, ix,iy;
        int t_y; // variable for head of time_total
                textattr(4);        // text is 4 RED
                textback; // background  2 is GREEN
                for (sy=5,iy=0;iy<4;iy++,sy=sy+4)

                        for (sx=8,ix=0;ix<4;ix++){
                                draw_border (sx, sy, sx+10, sy+2);
                                // start_x, start_yendx, end_x, end_y
                                sx = 18+sx;
                                //draw_border (sx, sy, sx+10, sy+2);
                                // start_x, start_yendx, end_x, end_y
                        }
                for (t_y=8;t_y<=20;t_y=t_y+4){
                        gotoxy(4,t_y);
                        cprintf("Tr:");
    }
                normvideo();
                return 1;
}
//void draw_border(int startx, int starty, int endx, int endy)

int w_on_block(val) // val : is start count of machine
int val;
{       int ic,ir;
        int ar_col[4] = { 10, 28, 46, 64 };//column position of machine-
                                                //block
        int ar_row[4] = {  6, 10, 14, 18 };
                // row : 6, 10, 14,18
```

```
                    // col : 10, 28, 46, 64
                    for (ir=0;ir<4;ir++)
                            for (ic=0;ic<4;ic++){
                              //gotoxy (ar_col[ic], row);  // x_pos, y_post
                              gotoxy (ar_col[ic],ar_row[ir]);
                              printf("%d",val++);
                             }
                    return 1;
}
// ***********************************************************
// PROCEDURE put_stat_on_block is putting the status of
// machine to block screen  (px:pos_of_x, py:pos_of_y,
// status:0=>off, 1=>on, 2=>no_used

int put_stat_on_block (px,py,status )
int px, py, status;
{       int ar_col[4] = { 10, 28, 46, 64 };
// column position of machine block
        int ar_row[4] = {  6, 10, 14, 18 };
// row position of machine block
                    gotoxy ( px+3, py);
                    if (status==0){
                            highvideo();
                            cprintf("OFF ");
                    }

                    if (status==1){
                            highvideo();
                            cprintf("ON ");
                    }
                    if (status==2){
                            highvideo();
                            cprintf(" - ");
                    }
                    return 1;

}
int put_time_total(px,py,n)
int px,py,n;
{
                    gotoxy(px,py);
                    printf("%5d:%2d",mach[n].PM_HR[0],mach[n].P_PM_HR[0]);
                    return 0;
}
int tool_mark()
{       int ix,iy,i;
                    ix = 1;
                    for (i=0;i<80;i++,ix++)
                    {
                            gotoxy(ix,2);
                            //if ((i==10)||(i==20)||(i==30))
                            if ((i%10)==0)
```

```
                        printf("X");
                else
                printf("þ");
        }
        iy = 1;
        for (i=0;i<25;i++,iy++)
        {       gotoxy(2,iy);
                if ((i%10)==0)
                        printf("X");
                else
                printf("þ");
        }
        return 1;
}
int curon()
{.      union REGS regs;
        regs.h.ch = 0x00;
        regs.h.cl = 13;
        regs.h.ah = 1;
        int86(0x10,&regs,&regs);
        return 1;
}
int curoff()-
{       union REGS regs;
        regs.h.ch = 0x20;
        regs.h.cl =0;
        regs.h.ah =1;
        int86(0x10,&regs,&regs);
        return 1;
}
int check_condition()
{       printf("THIS PROCESS IS THE FIRST RUNNING\n");
        printf("Press 'y'or 'Y' for 'YES' and press any key for 'NO'\n");
        ch = getch();
        if ((ch=='y')||(ch=='Y')){

                return 0;
        }
        else
                return 1;
}
void G_main(void)
{
        int mode,driver;
        int errorcode;
  int c;
  int r;
        int i,j;

                driver = DETECT;
                initgraph(&driver,&mode,"");
                errorcode = graphresult();
```

```
                maxx = getmaxx();
                maxy = getmaxy();
    c = r = 0;
                if(errorcode != grOk){
                printf("Graphic error:%s\n",grapherrormsg(errorcode));
                exit(-1);
                }
                Screen(0,0,maxx,maxy,txt);

                Table(0,0);   // 1

                Table(1,0);   // 2

                Table(2,0);   // 3

                Table(3,0);   // 4

                Table(0,1);   // 5

                Table(1,1);   // 6

                Table(2,1);   // 7

                Table(3,1);   // 8

                S_stus(0);
}
void Screen(int x1,int y1,int x2,int y2,char *str)
{
int l,x,h,i,j,k;
char date[20];
struct date d;

                setfillstyle(SOLID_FILL,LIGHTGRAY);
                setlinestyle(SOLID_LINE,0,WHITE);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
                bar(x1,y1,x2,y2);
                setcolor(BLACK);
                moveto(x1,y1);
                lineto(x2,y1);
                lineto(x2,y2);
                moveto(x1+1,y1+1);
                lineto(x2-1,y1+1);
                lineto(x2-1,y2-1);
                setcolor(WHITE);
                moveto(x1,y1);
                lineto(x1,y2);
                lineto(x2,y2);
                moveto(x1+1,y1+1);
                lineto(x1+1,y2-1);
                lineto(x2-1,y2-1);
                x = (x2-x1) / 2;
```

```
                x = x1+x;
                l = textwidth(str);
                h = textheight(str);
                setfillstyle(SOLID_FILL,BLUE);
                Vibar(x-(l/2)-5,y1+5,x+(l/2),y1+16+h);
                setcolor(WHITE);
                outtextxy(x-(l / 2),y1+10,str);
                j = 0;
                k = 0;
                settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
                getdate(&d);
                sprintf(date,"%2d:%2d:%4d",d.da_day,d.da_mon,d.da_year);
                for( i = 0 ; i < 4; i++){
                                k = j+7;
                                j += 157;
                                Vbar(x1+k,y1+45,x1+j,y1+45+20);
                                setcolor(WHITE);
                                outtextxy(x1+k+10,y1+52,fkey[i]);
                        }
   setcolor(LIGHTGREEN);
            outtextxy(x1+67,y1+52,date);
            setfillstyle(SOLID_FILL,BLACK);
   Vbar(x1+5,y1+75,x2-5,y2-85);//put black rectang to mid of screen
   setfillstyle(SOLID_FILL,BLACK);
   Vbar(x1+5,y2-70,x2-5,y2-5);
                return;
}
void Vbar(x1,y1,x2,y2)
{
  bar(x1,y1,x2,y2);
          setcolor(WHITE);
          moveto(x1,y1);
          lineto(x1,y2);
          lineto(x2,y2);
          setcolor(BLACK);
          lineto(x2,y1);
          lineto(x1+1,y1);
}
void Vibar(x1,y1,x2,y2)
{
  bar(x1,y1,x2,y2);
          setcolor(BLACK);
          moveto(x1,y1);
          lineto(x1,y2);
          lineto(x2,y2);
          setcolor(WHITE);
          lineto(x2,y1);
          lineto(x1,y1);
}
void Table(int r,int c)  //,int stus)
{
int i,j,x,y;
```

```
        x = 35;
        for( j = 0;j < 4;j++){
                y = 0;
                for( i = 0;i < 2;i++){
                        y += 130;
                        if(j == r && i == c){
                                table(x,y);//,stus);
                        }else;
                }
                x += 150;
        }
}
void table(int x,int y)//,int stus)
{
char sBuf[8];
        setfillstyle(SOLID_FILL,LIGHTGRAY);
        bar(x,y,x+120,y+70);
        setfillstyle(SOLID_FILL,BLUE);
        Vbar(x+10,y+10,x+110,y+30);
        Vbar(x+10,y+40,x+110,y+60);
        return;
}
void OutTime(int r,int c)
{
int x,y,i,j,z;
char tBuf[20];
struct time t;

        x = 35;
        z = 0;

        for( j = 0;j < 4;j++){
                y = 0;
                for( i = 0;i < 2;i++){
                        y += 130;
                        if(j == r && i == c){
                                gettime(&t);
                                sprintf(tBuf,time_o[z]);

        Vbar(x+10,y+40,x+110,y+60);
                                setcolor(LIGHTGREEN);
                                outtextxy(x+30,y+47,tBuf);
                        }else
                                z++;

                }
                x += 150;
        }
        return;
}
void Out_STime(void)
{
char tBuf[20];
```

```
struct time t;

        gettime(&t);
        sprintf(tBuf,"%2d:%2d:%2d",t.ti_hour,t.ti_min,t.ti_sec);
        Vbar(164,45,315,65);
        setcolor(WHITE);
        outtextxy(175,52,"TIME:");
        setcolor(LIGHTGREEN);
        outtextxy(230,52,tBuf);
        return;
}
void S_stus(int st)
{
        Vbar(478,45,628,45+20);
  setcolor(WHITE);
        outtextxy(478+10,52,"STATUS:");
        if(st){
                setcolor(LIGHTRED);
                outtextxy(478+90,52,status[1]);
        }else if(!st){
                setcolor(LIGHTGREEN);
                outtextxy(478+90,52,status[0]);
        }
  return;
}
void Out_hour(int hr,int min,int sec)
{
int hBuf[10];

        Vbar(321,45,471,45+20);
        setcolor(WHITE);
        sprintf(hBuf,"%d:%d:%d",hr,min,sec);
        outtextxy(321+10,52,"HOUR:");
        setcolor(LIGHTGREEN);
        outtextxy(321+70,52,hBuf);
}

void SetTime(int r,int c,int h,int m,int s)
{
int x = 0;
int i,j;
        for( j = 0;j < 4;j++){
                for( i = 0;i < 2;i++){
                                if(j == r && i == c){
                                  sprintf(time_o[x],"%d:%d:%d",h,m,s);
                                }else;
                x++;
                }
        }
}
void txt_Table(int r, int c, int stus)
                //Table(int r, int c,int stus)
```

```
{
int i,j,x,y;

        x = 35;
        for( j = 0;j < 4;j++){
                y = 0;
                for( i = 0;i < 2;i++){
                        y += 130;
                        if(j == r && i == c){
                                txt_table(x,y,stus);
                        }else;
                }
                x += 150;
        }
}
void txt_table(int x, int y, int stus)
                //table(int x, int y, int stus)
{

char sBuf[8];

        setfillstyle(SOLID_FILL,LIGHTGRAY);
        //bar(x,y, // block of machine
        setfillstyle(SOLID_FILL,BLUE);

        Vbar(x+10,y+10,x+110,y+30); // Bar of upper text

        if( stus==0 ){
                sprintf(sBuf,"OFF");
                setcolor(LIGHTRED);
                outtextxy(x+50,y+17,sBuf);
        }else
        if( stus==1 ){
                sprintf(sBuf, "ON");
                setcolor(LIGHTGREEN);
                outtextxy(x+50,y+17,sBuf);
        }else
        if( stus==2) {
                sprintf(sBuf, " -");
                setcolor(WHITE);
                outtextxy(x+50,y+17,sBuf);
        }else
        return;
}
/* INTERRUPT SERVICE ROUTINE */

void interrupt Clock()
{
 disable() ;
 clkcount++ ;

                if((clkcount > 17)&&(Xck_==1))
```

```
                {       clkcount = 0 ;
                        tick = 1;
                        xc = xc+1;
                }
                if (xc==4){
                        Xck_ = 0;
                        xc = 0;
                }
                if((clkcount > 18)&&(Xck_==0))
                {       clkcount = 0;
                        tick = 1;
                        Xck_ = 1;
                }
        enable() ;
}

void interrupt Com2()
{
        disable() ;
        *(vid_mem+6) = c2 ;
        outportb(0x300, 0000) ;
        *(vid_mem+8) = c2++ ;
        outportb(0x20, 0x20) ;
        outportb(0x21, mask&0x7f) ;
        enable();
}

void interrupt Com1()
{
        disable() ;
        outportb(0x300, 0x00) ;
        *(vid_mem+2) = '1' ;
        outportb(0x300, 0x00) ;
        *(vid_mem+4) = '1' ;
        enable() ;
        asm sti ;
}
void InitPort(void)
{
        outportb(0x330, 0x0) ;

}

int vid_mode()
{
 union REGS r;

 r.h.ah = 15;
 return int86(0x10, &r, &r) & 255;
}

void INT_main()
```

```
{
        struct time t;
        int mode ;
        int i,x,sum,SUM;
        char ch;
        SUM = 0;
InitPort();
        clrscr() ;
        outportb(0x20, 0x20) ;
        mask = inportb(0x21);
        outportb(0x21, mask&0x7f) ;
        OldClock = getvect(TIMER) ;
        OldCom2 = getvect(COM2) ;
        setvect(TIMER, Clock) ;
        setvect(IRQ7, Com2) ;
        x = sum = 0;
        tick = 1;
        do{
                gettime(&t);
                gotoxy(10,9);
                printf("t.ti_sec= %02d",t.ti_sec);
        }while(t.ti_sec!=59);
        SUM = t.ti_min;
        sum = t.ti_sec;
        clrscr();
        while(!kbhit())
        {
                gotoxy(10,10);
                gettime(&t);
                printf("time min:%02d sec:%02d \n",t.ti_min,t.ti_sec);
                gotoxy(10,11);
                if (tick==1)
                {
                        x = 1;
                        sum = sum+1;
                        tick = 0;
                }
                else
                        x = 0;
                if (sum>59)
                {       sum = 0;
                        SUM = SUM+1;
                }
                printf("SUM =   :%02d sum:%02d \n",SUM,sum);
                delay(800);
        }
        printf(" After the while Routine \n");
        setvect(TIMER, OldClock) ;
        setvect(COM2, OldCom2) ;
        outportb(0x21, mask) ;
        printf("\n\n\nOld Interrupt enable is %2X\n", mask) ;
        printf("\nSet Interrupt enable to %2X\n", mask) ;
```

```
        getch() ;

}
void edit_change(void)
{
        int ei,n,no_1,no_2;
        unsigned int ehr;
        char e_ch;
        clrscr();
        do{
          gotoxy(1,1);
          printf("                ");
          gotoxy(1,1);
          printf("Enter Machine No.:");
          scanf("%d",&n);
        }while(n>8);
        no_1 = n-1;
        printf("machine name:%s\n",mach[no_1].N_mach);
        printf("Maximum check-point is %d\n",mach[no_1].act_cp);
        for(ei=0;ei<mach[no_1].act_cp;ei++)
        {  printf("STATUS of CHECK-POINT %d ",ei+1);
           if(mach[no_1].chang[ei]==0)
             printf(" CHANGE O.K. :%d count\n",mach[no_1].chang[ei]);
           else
             printf(" NOT CHANGE  :%d count\n",mach[no_1].chang[ei]);

        }
        printf("\nWhat the CHECK-POINT to change ? :");
        scanf("%d",&n);
        no_2 = n-1;
        printf("CHECK-POINT %d status of change :",no_2+1);
        if(mach[no_1].chang[no_2]==0)
                printf(" CHANGE O.K. :%d count\n",mach[no_1].chang[no_2]);
        else
                printf(" NOT CHANGE  :%d count\n",mach[no_1].chang[no_2]);

        printf("Are you sure to change? <y/n>");
        e_ch = getch();
        e_ch = tolower(e_ch);
        if(e_ch=='y'){

                mach[no_1].chang[no_2] = 0;
                ehr = mach[no_1].P_PM_HR[no_2] + mach[no_1].P_Freq[no_2];
    if(ehr >= 60 )
    { mach[no_1].NEXTPM[no_2] = mach[no_1].NEXTPM[no_2]+1;
      mach[no_1].P_NEXTPM[no_2] = mach[no_1].P_PM_HR[no_2] +
      mach[no_1].P_Freq[no_2] - 60;

    }
     else{
      mach[no_1].P_NEXTPM[no_2] = mach[no_1].P_PM_HR[no_2] +
      mach[no_1].P_Freq[no_2];
```

```
        mach[no_1].NEXTPM[no_2] = mach[no_1].PM_HR[no_2] +
        mach[no_1].Freq[no_2];
        }
        /*mach[no_1].P_NEXTPM[no_2] = mach[no_1].P_PM_HR[no_2] +
         mach[no_1].P_Freq[no_2];
                        mach[no_1].NEXTPM[no_2] = mach[no_1].PM_HR[no_2] +
                         mach[no_1].Freq[no_2];
                        OLD IS NO BRACKET*/
                        SAVE_monthly_FINISHED(no_1, no_2);
                }
                printf("\nCHECK-POINT %d status of change :",no_2+1);
                if(mach[no_1].chang[no_2]==0)
                        printf(" CHANGE O.K. :%d count\n",mach[no_1].chang[no_2]);
                else
                        printf(" NOT CHANGE  :%d count\n",mach[no_1].chang[no_2]);
                //printf("\nSTATUS of CHANGE :%d \n",mach[no_1].chang[no_2]);
                SAVE_data_to_disk();
}
int SAVE_monthly_FINISHED(int M1, int M2)
{       unsigned int Mx,i,file_h;
        FILE *fpM,*fpR;
        char file_name_M[10], str_M[30];
        char MM_string[10];
        char ch_s[5];
                //printf("File name:  ");
                //gets(file_name);
                clear_ch_string();
                for(i=0;i<30;i++)
                        str_M[i] = '\0';
                if ((fpR = fopen("MONTH.REP","r"))==NULL){
                        file_h = 0;
                }
                else
                        file_h = 1;
                fclose(fpR);
                if(file_h == 0)
                        strcpy(ch_s,"w");
                else

                        strcpy(ch_s,"a");
                if ((fpM = fopen("MONTH.REP",ch_s))==NULL){
                        printf("Error in open file\n");
                        exit(1);
                }
                strcpy(ch_string, d_string);
                strcat(ch_string, "MACHINE NAME:");
                strcat(ch_string, mach[M1].N_mach);
                strcat(ch_string, " | LUB.CODE:");
                for(i=0;i<30;i++)
                        str_M[i] = '\0';
                strcat(ch_string, mach[M1].Lub_code[M2]);
                strcat(ch_string, " | LAST_PM:");
```

```
                for(i=0;i<30;i++)
                        str_M[i] = '\0';
                itoa(mach[M1].PM_LAST[M2], str_M,10);
                strcat(ch_string, str_M);
                for(i=0;i<30;i++)
                        str_M[i] = '\0';
                itoa(mach[M1].P_PM_LAST[M2], str_M, 10);
                strcat(ch_string, ":");
                strcat(ch_string, str_M);
                strcat(ch_string, "\n");
                fputs(ch_string,fpM);
                //printf("Enter string\n");
                //gets(str_M);
                /*
                while(*str != '\0') {
                        strcat(str,"\n");
                        fputs(str,fpM);
                        if(ferror(fpM)){
                                printf("Error in writing file\n");
                                exit(1);
                        }
                        gets(str_M);
                }
                printf("\007");
                */
                fclose(fpM);
                return 1;
}

main()
{       int i,cp;
        char m_ch;
                clrscr();
                condition_f = check_condition();
                //G_main();
                clear_data_mach();
                read_datafile();
                initial_port();
                //-------------------------
                InitPort();// for INTERUPT ROUTINE
                outportb(0x20, 0x20) ;
                mask = inportb(0x21);
                outportb(0x21, mask&0x7f) ;
                OldClock = getvect(TIMER) ;
                OldCom2 = getvect(COM2) ;
                setvect(TIMER, Clock) ;
                setvect(IRQ7, Com2) ;
                tick = 1;
                m_ch = 'a';
                p_warn = 1;
                p_Month = 1;
```

```
do{
        G_main();
        G_process_real(0);
        closegraph();
        edit_change();
        read_poons_r();
        clrscr();
        gotoxy(25,11);
        printf("Exit to DOS, Press 'x' or 'X'\n");
        gotoxy(20,12);
        printf("Return to Process Again press 'ENTER'");
        m_ch = getch();
        m_ch = tolower(m_ch);
}while (m_ch!='x');

setvect(TIMER, OldClock) ;
setvect(COM2, OldCom2) ;
outportb(0x21, mask) ;
return 1;
}
```

24.057

# ภาคผนวก ง

## แสดงผลจากการทดลองพิมพ์ใบสั่งงานสำหรับเครื่องจักรทั้ง 3 เครื่อง

## ภาคผนวก ง

## แสดงผลจากการทดลองพิมพ์ใบสั่งงานสำหรับเครื่องจักรทั้ง 3 เครื่อง

CHECK POINT :4

LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:21:47 1995

Standard Time:.....

Operating:

Lub. Code ---)C-MA-1-09

M/C Name ---)COATING

Part Name ---)MOTOR GEAR

Part Spec. ---)LIFT

Lubricant ---)OIL

Grade ---)MOBIL GEAR 630

Method ---)REPLACE

Quantity ---)1.2 LIT.

Frequency ---)0:6

Pm. Hour No.---)0:6

Results:

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :4

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:29:43 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-09

M/C Name ---)COATING

Part Name ---)MOTOR GEAR

Part Spec. ---)LIFT

Lubricant ---)OIL

Grade ---)MOBIL GEAR 630

Method ---)REPLACE

Quantity ---)1.2 LIT.

Frequency ---)0:6

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer ih order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :5

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:26:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-10

M/C Name ---)COATING

Part Name ---)GEAR BOX

Part Spec. ---)FEEDING

Lubricant ---)OIL

Grade ---)MOBIL GEAR 630

Method ---)REPLACE

Quantity ---)1.2 LIT.

Frequency ---)0:9

Pm. Hour No.---)0:9

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....



CHECK POINT :6

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:26:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-24

M/C Name ---)COATING

Part Name ---)AIR COMPRESSOR

Part Spec. ---)FEEDING

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)60 g.

Frequency ---)0:9

Pm. Hour No.---)0:9

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :9

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:20:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-33

M/C Name ---)COATING

Part Name ---)CHAIN

Part Spec. ---)SHEET PUSHER

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)100 g.

Frequency ---)0:5

Pm. Hour No.---)0:5

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :9

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:27:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-33

M/C Name ---)COATING

Part Name ---)CHAIN

Part Spec. ---)SHEET PUSHER

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)100 g.

Frequency ---)0:5

Pm. Hour No.---)0:10

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :11

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:27:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-35

M/C Name ---)COATING

Part Name ---)CHAIN

Part Spec. ---)STOPPER

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)60 g.

Frequency ---)0:10

Pm. Hour No.---)0:10

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :12

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:18:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-36

M/C Name ---)COATING

Part Name ---)CHAIN & SPROCKET

Part Spec. ---)FEEDING TABLE

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:3

Pm. Hour No.---)0:3

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :12

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:21:47 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-36

M/C Name ---)COATING

Part Name ---)CHAIN & SPROCKET

Part Spec. ---)FEEDING TABLE

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:3

Pm. Hour No.---)0:6

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :12

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:26:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-36

M/C Name ---)COATING

Part Name ---)CHAIN & SPROCKET

Part Spec. ---)FEEDING TABLE

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:3

Pm. Hour No.---)0:9

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :12

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:29:43 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-36

M/C Name ---)COATING

Part Name ---)CHAIN & SPROCKET

Part Spec. ---)FEEDING TABLE

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:3

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :15

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:24:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-45

M/C Name ---)COATING

Part Name ---)GEAR

Part Spec. ---)MAIN MOTOR

Lubricant ---)OIL

Grade ---)MOBIL GEAR 630

Method ---)REPLACE

Quantity ---)1.2 LIT.

Frequency ---)0:7

Pm. Hour No.---)0:7

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :16

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:21:47 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-51

M/C Name ---)COATING

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)COATING 10B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:6

Pm. Hour No.---)0:6

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :16

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:29:43 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-51

M/C Name ---)COATING

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)COATING 10B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:6

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :18

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:25:40 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-53

M/C Name ---)COATING

Part Name ---)GEAR REDUCER

Part Spec. ---)SYNCHRONIZE

Lubricant ---)OIL

Grade ---)SHELL OMALA 680

Method ---)REPLACE

Quantity ---)0.5 LIT.

Frequency ---)0:8

Pm. Hour No.---)0:8

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :19

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:28:43 1995

Standard Time:.....

**Operating:**

Lub. Code ---)C-MA-1-54

M/C Name ---)COATING

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)SYNCHRONIZE 10B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:11

Pm. Hour No.---)0:11

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :1

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:25:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-01

M/C Name ---)SCROLL SHEAR

Part Name ---)HYDRAULIC PUMP

Part Spec. ---)FEEDING

Lubricant ---)OIL

Grade ---)MOBIL DTE 26

Method ---)REPLACE

Quantity ---)5 LIT.

Frequency ---)0:8

Pm. Hour No.---)0:8

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :3

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:23:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-07

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)05B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)20 g.

Frequency ---)0:6

Pm. Hour No.---)0:6

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :3

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:29:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-07

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)05B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)20 g.

Frequency ---)0:6

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :4

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:21:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-10

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHIAN

Part Spec. ---)10B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:4

Pm. Hour No.---)0:4

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :4

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:25:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-10

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHIAN

Part Spec. ---)10B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:4

Pm. Hour No.---)0:8

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :4

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:29:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-10

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHIAN

Part Spec. ---)10B

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:4

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :5

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:21:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-11

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)10B REDUCER

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:4

Pm. Hour No.---)0:4

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :5

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:25:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-11

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)10B REDUCER

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:4

Pm. Hour No.---)0:8

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :5

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:29:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-11

M/C Name ---)SCROLL SHEAR

Part Name ---)SPROCKET & CHAIN

Part Spec. ---)10B REDUCER

Lubricant ---)GREASE

Grade ---)SHELL R3

Method ---)TOP-UP

Quantity ---)40 g.

Frequency ---)0:4

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :9

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:24:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-17

M/C Name ---)SCROLL SHEAR

Part Name ---)MOTOR GEAR

Part Spec. ---)SCROLL SHEAR

Lubricant ---)OIL

Grade ---)MOBIL GEAR 630

Method ---)REPLACE

Quantity ---)0.5 LIT.

Frequency ---)0:7

Pm. Hour No.---)0:7

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :9

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:31:15 1995

Standard Time:.....

**Operating:**

Lub. Code ---)S-KG-2-17

M/C Name ---)SCROLL SHEAR

Part Name ---)MOTOR GEAR

Part Spec. ---)SCROLL SHEAR

Lubricant ---)OIL

Grade ---)MOBIL GEAR 630

Method ---)REPLACE

Quantity ---)0.5 LIT.

Frequency ---)0:7

Pm. Hour No.---)0:14

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :2

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:23:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)BBB

M/C Name ---)TESTING

Part Name ---)BBB

Part Spec. ---)BBB

Lubricant ---)BBB

Grade ---)BBB

Method ---)BBB

Quantity ---)BBB

Frequency ---)0:8

Pm. Hour No.---)0:8

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :3

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:19:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)CCC

M/C Name ---)TESTING

Part Name ---)CCC

Part Spec. ---)CCC

Lubricant ---)CCC

Grade ---)CCC

Method ---)CCC

Quantity ---)CCC

Frequency ---)0:4

Pm. Hour No.---)0:4

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :3

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:23:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)CCC

M/C Name ---)TESTING

Part Name ---)CCC

Part Spec. ---)CCC

Lubricant ---)CCC

Grade ---)CCC

Method ---)CCC

Quantity ---)CCC

Frequency ---)0:4

Pm. Hour No.---)0:8

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :3

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:27:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)CCC

M/C Name ---)TESTING

Part Name ---)CCC

Part Spec. ---)CCC

Lubricant ---)CCC

Grade ---)CCC

Method ---)CCC

Quantity ---)CCC

Frequency ---)0:4

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :5

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:21:47 1995

Standard Time:.....

**Operating:**

Lub. Code ---)EEE

M/C Name ---)TESTING

Part Name ---)EEE

Part Spec. ---)EEE

Lubricant ---)EEE

Grade ---)EEE

Method ---)EEE

Quantity ---)EEE

Frequency ---)0:6

Pm. Hour No.---)0:6

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

CHECK POINT :5

# LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD.

CC:ENGINEERING DIVISION

Printed Date: Thu Jan 12 21:27:48 1995

Standard Time:.....

**Operating:**

Lub. Code ---)EEE

M/C Name ---)TESTING

Part Name ---)EEE

Part Spec. ---)EEE

Lubricant ---)EEE

Grade ---)EEE

Method ---)EEE

Quantity ---)EEE

Frequency ---)0:6

Pm. Hour No.---)0:12

**Results:**

[ ] Complete

.....

[ ] Comment

.....

.....

.....

DATE:.....

Started time:.....

Finished time:.....

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

# ภาคผนวก จ

## แสดงผลจากการทดลองพิมพ์ใบสั่งงานที่ยังไม่ทำ(WARNING)

## ภาคผนวก จ

### แสดงผลจากการทดลองพิมพ์ใบสั่งงานที่ยังไม่ได้ทำ (WARNING)

** WARNING **

CHECK POINT :4

LUBRICATING WORK ORDER

POONSUB CAN CO.,LTD. CC:ENGINEERING DIVISION

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

**Operating:** **Results:**

Lub. Code —›C-MA-1-09 [ ] Complete

M/C Name —›COATING .....

Part Name —›MOTOR GEAR [ ] Comment

Part Spec. —›LIFT .....

Lubricant —›OIL .....

Grade —›MOBIL GEAR 630 .....

Method —›REPLACE DATE:.....

Quantity —›1.2 LIT. Started time:.....

Frequency —›0:6 Finished time:.....

Pm. Hour No.—›0:13

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :6

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

**Operating:** **Results:**

Lub. Code —›C-MA-1-24 [ ] Complete

M/C Name —›COATING .....

Part Name —›AIR COMPRESSOR[ ] Comment

Part Spec. —›FEEDING .....

Lubricant —›GREASE .....

Grade —›SHELL R3 .....

Method —›TOP-UP DATE:.....

Quantity —›60 g. Started time:.....

Frequency —›0:9 Finished time:.....

Pm. Hour No.—›0:13

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :9

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

**Operating:** **Results:**

Lub. Code —›C-MA-1-33 [ ] Complete

M/C Name —›COATING .....

Part Name —›CHAIN [ ] Comment

Part Spec. —›SHEET PUSHER .....

Lubricant —›GREASE .....

Grade —›SHELL R3 .....

Method —›TOP-UP DATE:.....

Quantity —›100 g. Started time:.....

Frequency —›0:5 Finished time:.....

Pm. Hour No.—›0:13

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :11

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

| **Operating:** | **Results:** |
|---|---|
| Lub. Code —›C-MA-1-35 | [ ] Complete |
| M/C Name —›COATING | ..... |
| Part Name —›CHAIN | [ ] Comment |
| Part Spec. —›STOPPER | ..... |
| Lubricant —›GREASE | ..... |
| Grade —›SHELL R3 | ..... |
| Method —›TOP-UP | DATE:..... |
| Quantity —›60 g. | Started time:..... |
| Frequency —›0:10 | Finished time:..... |
| Pm. Hour No.—›0:13 | |

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :18

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

**Operating:** **Results:**

Lub. Code —›C-MA-1-53 [ ] Complete

M/C Name —›COATING .....

Part Name —›GEAR REDUCER [ ] Comment

Part Spec. —›SYNCHRONIZE .....

Lubricant —›OIL .....

Grade —›SHELL OMALA 680.....

Method —›REPLACE DATE:.....

Quantity —›0.5 LIT. Started time:.....

Frequency —›0:8 Finished time:.....

Pm. Hour No.—›0:13

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :1

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

**Operating:** **Results:**

Lub. Code —›S-KG-2-01 [ ] Complete

M/C Name —›SCROLL SHEAR .....

Part Name —›HYDRAULIC PUMP [ ] Comment

Part Spec. —›FEEDING .....

Lubricant —›OIL .....

Grade —›MOBIL DTE 26 .....

Method —›REPLACE DATE:.....

Quantity —›5 LIT. Started time:.....

Frequency —›0:8 Finished time:.....

Pm. Hour No.—›0:14

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :4

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

**Operating:** **Results:**

Lub. Code ---›S-KG-2-10 [ ] Complete

M/C Name ---›SCROLL SHEAR .....

Part Name ---›SPROCKET & CHIAN] Comment

Part Spec. ---›10B .....

Lubricant ---›GREASE .....

Grade ---›SHELL R3 .....

Method ---›TOP-UP DATE:.....

Quantity ---›40 g. Started time:.....

Frequency ---›0:4 Finished time:.....

Pm. Hour No.---›0:14

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

** WARNING **

CHECK POINT :5

# LUBRICATING WORK ORDER

**POONSUB CAN CO.,LTD.** **CC:ENGINEERING DIVISION**

Printed Date: Tue Jan 17 00:00:00 1995

Standard Time:.....

| **Operating:** | **Results:** |
|---|---|
| Lub. Code ---›EEE | [ ] Complete |
| M/C Name ---›TESTING | ..... |
| Part Name ---›EEE | [ ] Comment |
| Part Spec. ---›EEE | ..... |
| Lubricant ---›EEE | ..... |
| Grade ---›EEE | ..... |
| Method ---›EEE | DATE:..... |
| Quantity ---›EEE _ _ | Started time:..... |
| Frequency ---›0:6 | Finished time:..... |
| Pm. Hour No.---›0:15 | |

SECTION CHIEF:..... PREPARED BY .....

PERFORMED BY .....

DIVISION MGR :..... ACKNOWLEDGED BY .....

[ ] RETURN TO ENGINEERING DIVISION

After this job finished, this data must be input to Computer in order

to update this lub.code for the next due. Entered by:.....

# ภาคผนวก ฉ

## แสดงข้อมูลในไฟล์หลัก (MASTER FILE) สำหรับเครื่องจักรทั้ง 3 เครื่อง

ภาคผนวก ฉ

## แสดงข้อมูลในไฟล์หลัก(MASTER FILE)สำหรับเครื่องจักรทั้ง 3 เครื่อง

DETAIL OF LUBRICATING

MACHINE NAME: COATING

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 1
3. LUBRICANT CODE : C-MA-1-01
4. PART NAME OF M/C : MOTOR GEAR
5. SPECIFICATION OF PART NAME : CONVEYOR(1)
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.5 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 2
3. LUBRICANT CODE : C-MA-1-02
4. PART NAME OF M/C : MOTOR GEAR
5. SPECIFICATION OF PART NAME : CONVEYOR(2)
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.5 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 3
3. LUBRICANT CODE : C-MA-1-07-08
4. PART NAME OF M/C : GEAR
5. SPECIFICATION OF PART NAME : LIFT L-R
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.8 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 4
3. LUBRICANT CODE : C-MA-1-09
4. PART NAME OF M/C : MOTOR GEAR
5. SPECIFICATION OF PART NAME : LIFT
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 1.2 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 5
3. LUBRICANT CODE : C-MA-1-10
4. PART NAME OF M/C : GEAR BOX
5. SPECIFICATION OF PART NAME : FEEDING
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 1.2 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 6
3. LUBRICANT CODE : C-MA-1-24
4. PART NAME OF M/C : AIR COMPRESSOR
5. SPECIFICATION OF PART NAME : FEEDING
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 60 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 7
3. LUBRICANT CODE : C-MA-1-25
4. PART NAME OF M/C : VACUUM PMP
5. SPECIFICATION OF PART NAME : FEEDING
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL DTE OIL AA
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 1.2 LIT.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 8
3. LUBRICANT CODE : C-MA-1-32
4. PART NAME OF M/C : SHEET PUSHER
5. SPECIFICATION OF PART NAME : FEEDING TABLE
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 9
3. LUBRICANT CODE : C-MA-1-33
4. PART NAME OF M/C : CHAIN
5. SPECIFICATION OF PART NAME : SHEET PUSHER
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 100 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95-
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 10
3. LUBRICANT CODE : C-MA-1-34
4. PART NAME OF M/C : SPROCKET
5. SPECIFICATION OF PART NAME : SHEET PUSHER
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHEEL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 11
3. LUBRICANT CODE : C-MA-1-35
4. PART NAME OF M/C : CHAIN
5. SPECIFICATION OF PART NAME : STOPPER
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 60 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 12
3. LUBRICANT CODE : C-MA-1-36
4. PART NAME OF M/C : CHAIN & SPROCKET
5. SPECIFICATION OF PART NAME : FEEDING TABLE
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 13
3. LUBRICANT CODE : C-MA-1-43
4. PART NAME OF M/C : GEAR REDUCER
5. SPECIFICATION OF PART NAME : COATING
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.2 LIT.
10. FREQUENCY HOUR : 3000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 14
3. LUBRICANT CODE : C-MA-1-44
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : COATING
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 15
3. LUBRICANT CODE : C-MA-1-45
4. PART NAME OF M/C : GEAR
5. SPECIFICATION OF PART NAME : MAIN MOTOR
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 1.2 LIT.
10. FREQUENCY HOUR : 3000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 16
3. LUBRICANT CODE : C-MA-1-51
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : COATING 108
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 17
3. LUBRICANT CODE : G-MA-1-52
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : COATING
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 60 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 18
3. LUBRICANT CODE : C-MA-1-53
4. PART NAME OF M/C : GEAR REDUCER
5. SPECIFICATION OF PART NAME : SYNCHRONIZE
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : SHELL OMALA 680
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.5 LIT.
10. FREQUENCY HOUR : 3000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 19
3. LUBRICANT CODE : C-MA-1-54
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : SYNCHRONIZE 10B
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 20
3. LUBRICANT CODE : C-MA-1-55
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : SYNCHRONIZE
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 21
3. LUBRICANT CODE : C-MA-1-56
4. PART NAME OF M/C : GEAR REDUCE
5. SPECIFICATION OF PART NAME : SYNCHRONIZE
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : SHELL OMALA 680
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.5 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 22
3. LUBRICANT CODE : C-MA-1-57
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : SYNCHRONIZE 06B
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 23
3. LUBRICANT CODE : C-MA-1-59
4. PART NAME OF M/C : GEAR BOX
5. SPECIFICATION OF PART NAME : SYNCHRONIZE
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : SHELL OMALA 680
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.8 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1¦MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 24
3. LUBRICANT CODE : C-MA-1-60
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : 12B
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 150 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 1 ¦ MACHINE NAME : COATING
2. MAX.POINT:25 ¦ CHECK_POINT : 25
3. LUBRICANT CODE : C-MA-1-48-49
4. PART NAME OF M/C : BRASS BUSH
5. SPECIFICATION OF PART NAME : COATING
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R&
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 60 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

# DETAIL OF LUBRICATING

# MACHINE NAME: SCROLL SHEAR

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 1
3. LUBRICANT CODE : S-KG-2-01
4. PART NAME OF M/C : HYDRAULIC PUMP
5. SPECIFICATION OF PART NAME : FEEDING
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL DTE 26
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 5 LIT.
10. FREQUENCY HOUR : 18000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 2
3. LUBRICANT CODE : S-KG-2-05
4. PART NAME OF M/C : VACUUM PUMP
5. SPECIFICATION OF PART NAME : FEEDING
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 60 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 3
3. LUBRICANT CODE : S-KG-2-07
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : 05B
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 20 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 4
3. LUBRICANT CODE : S-KG-2-10
4. PART NAME OF M/C : SPROCKET & CHIAN
5. SPECIFICATION OF PART NAME : 10B
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 5
3. LUBRICANT CODE : S-KG-2-11
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : 10B REDUCER
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 6
3. LUBRICANT CODE : S-KG-2-12
4. PART NAME OF M/C : DUPBAL BEARING
5. SPECIFICATION OF PART NAME : FEEDING TABLE
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 40 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 7
3. LUBRICANT CODE : S-KG-2-13
4. PART NAME OF M/C : MOTOR GEAR
5. SPECIFICATION OF PART NAME : MAIN
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 1.5 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 8
3. LUBRICANT CODE : S-KG-2-16
4. PART NAME OF M/C : SPROCKET & CHAIN
5. SPECIFICATION OF PART NAME : 10B
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 60 g.
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 9
3. LUBRICANT CODE : S-KG-2-17
4. PART NAME OF M/C : MOTOR GEAR
5. SPECIFICATION OF PART NAME : SCROLL SHEAR
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : REPLACE
9. QUANTITY OF LUBRICANT : 0.5 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 10
3. LUBRICANT CODE : S-KG-2-18
4. PART NAME OF M/C : BEARING
5. SPECIFICATION OF PART NAME : SHAFT OF BELT
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 20 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 11
3. LUBRICANT CODE : S-KH-2-24
4. PART NAME OF M/C : AIR CYLINDER
5. SPECIFICATION OF PART NAME : TP. COLLECTING
6. KIND OF LUBRICANT : GREASE
7. GRADE OF LUBRICANT : SHELL R3
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 20 g.
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 2¦MACHINE NAME : SCROLL SHEAR
2. MAX.POINT:12 ¦ CHECK_POINT : 12
3. LUBRICANT CODE : S-KG-2-25
4. PART NAME OF M/C : MOTOR GEAR
5. SPECIFICATION OF PART NAME : CONVEYOR
6. KIND OF LUBRICANT : OIL
7. GRADE OF LUBRICANT : MOBIL GEAR 630
8. METHOD OF LUBRICANT : TOP-UP
9. QUANTITY OF LUBRICANT : 0.5 LIT.
10. FREQUENCY HOUR : 6000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

# DETAIL OF LUBRICATING

# MACHINE NAME: TESTING

1. MACHINE NO. 3¦MACHINE NAME : TESTING
2. MAX.POINT: 5 ¦ CHECK_POINT : 1
3. LUBRICANT CODE : AAA
4. PART NAME OF M/C : AAA
5. SPECIFICATION OF PART NAME : AAA
6. KIND OF LUBRICANT : AAA
7. GRADE OF LUBRICANT : AAA
8. METHOD OF LUBRICANT : AAA
9. QUANTITY OF LUBRICANT : AAA
10. FREQUENCY HOUR : 500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 3¦MACHINE NAME : TESTING
2. MAX.POINT: 5 ¦ CHECK_POINT : 2
3. LUBRICANT CODE : BBB
4. PART NAME OF M/C : BBB
5. SPECIFICATION OF PART NAME : BBB
6. KIND OF LUBRICANT : BBB
7. GRADE OF LUBRICANT : BBB
8. METHOD OF LUBRICANT : BBB
9. QUANTITY OF LUBRICANT : BBB
10. FREQUENCY HOUR : 1000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 3¦MACHINE NAME : TESTING
2. MAX.POINT: 5 ¦ CHECK_POINT : 3
3. LUBRICANT CODE : CCC
4. PART NAME OF M/C : CCC
5. SPECIFICATION OF PART NAME : CCC
6. KIND OF LUBRICANT : CCC
7. GRADE OF LUBRICANT : CCC
8. METHOD OF LUBRICANT : CCC
9. QUANTITY OF LUBRICANT : CCC
10. FREQUENCY HOUR : 1500: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 3¦MACHINE NAME : TESTING
2. MAX.POINT: 5 ¦ CHECK_POINT : 4
3. LUBRICANT CODE : DDD
4. PART NAME OF M/C : DDD
5. SPECIFICATION OF PART NAME : DDD
6. KIND OF LUBRICANT : DDD
7. GRADE OF LUBRICANT : DDD
8. METHOD OF LUBRICANT : DDD
9. QUANTITY OF LUBRICANT : DDD
10. FREQUENCY HOUR : 2000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 7/1/95
15. DONE BY SECTION : PM.

---

1. MACHINE NO. 3¦MACHINE NAME : TESTING
2. MAX.POINT: 5 ¦ CHECK_POINT : 5
3. LUBRICANT CODE : EEE
4. PART NAME OF M/C : EEE
5. SPECIFICATION OF PART NAME : EEE
6. KIND OF LUBRICANT : EEE
7. GRADE OF LUBRICANT : EEE
8. METHOD OF LUBRICANT : EEE
9. QUANTITY OF LUBRICANT : EEE
10. FREQUENCY HOUR : 3000: 0
11. PM.LAST HOUR : 0: 0
12. PM. HOUR : 0: 0: 0
13. NEXT PM. HOUR : 0: 0
14. DD / MM / YY : 0
15. DONE BY SECTION : PM.

---

# ประวัติผู้เขียน

นายศิริพงษ์ ม่วงศิริ เกิดวันที่ 7 มกราคม 2505 ที่อำเภอบางขุนเทียน กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตรบัณฑิต สาขาฟิสิกส์ คณะ - วิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า ธนบุรี ในปีการศึกษา 2525 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ.2535 ปัจจุบันทำงานที่ บริษัทพูนทรัพย์แคน จำกัด อำเภอเมือง จังหวัดสมุทรสาคร 74000 .