

# โปรแกรมแสดงแผนที่

## Electronic Map



นายฉัตรชัย ใจเข้มแข็ง รหัส 41360256  
นายวุฒิภูมิ อกนิษฐ์ รหัส 41360439

ท้องถนคณบดีวิทยากรรมศาสตร์
วันที่รับ... 3,0 พ.ย. 2544
เลขทะเบียน... ๑๓. 44005๑4
เลขเรียกหนังสือ... ๘
มหาวิทยาลัยนครสวรรค์ ๒๓๖.๖, ๓๒๑๓/

1509776X  
รศ.  
ณ.๒๒๓  
๒๕๔๔

ปริญญานิพนธ์เป็นส่วนหนึ่งของหลักสูตรปริญญาวิทยาศาสตรบัณฑิต ๒๕๔๔ C.1

สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์

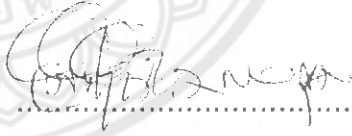
ปีการศึกษา 2544

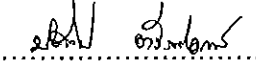


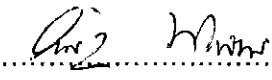
## ใบรับรองโครงการวิจัย

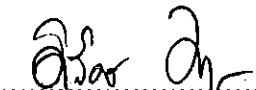
หัวข้อโครงการ โปแกรมแสดงแผนที่  
ผู้ดำเนินโครงการ นายฉัตรชัย ใจเข้มแข็ง รหัส 41360256  
นายวุฒิภูมิ อกนิษฐ์ รหัส 41360439  
อาจารย์ที่ปรึกษา อาจารย์สถิตินิโชค เชาวกุล  
สาขา วิศวกรรมคอมพิวเตอร์  
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา 2544

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการ  
ศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์  
คณะกรรมการสอบโครงการวิจัย

  
.....ประธานกรรมการ  
(อาจารย์สถิตินิโชค เชาวกุล)

  
.....กรรมการ  
(อาจารย์ประทีป ศิริรัตนโสภาส)

  
.....กรรมการ  
(อาจารย์วัชรวิทย์ พิษพันธ์)

  
.....กรรมการ  
(อาจารย์ศิริพร เดชะศิริรักษ์)

หัวข้อโครงการ : โปรแกรมแสดงแผนที่  
ผู้ดำเนินโครงการ : นายฉัตรชัย ใจเข้มแข็ง รหัส 41360256  
~~นายวุฒิกุมิ อกนิษฐ์ รหัส 41360439~~  
อาจารย์ที่ปรึกษา : อาจารย์สิทธิโชค เซาวกุล  
สาขา : วิศวกรรมคอมพิวเตอร์  
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา : 2544

#### บทคัดย่อ

โครงการนี้เป็นการศึกษาและพัฒนาโปรแกรมสำหรับใช้เพื่อเป็นประโยชน์ในการศึกษาเส้นทางจราจร โดยได้จัดทำเป็นรูปแบบโปรแกรมสำเร็จรูป ใช้แสดงเส้นทางจราจรของภาคเหนือตอนล่าง ซึ่งลักษณะของโปรแกรมที่ใช้จะใช้รูปแบบของภาษาวิซวลเบสิก เวอร์ชัน 6.0 (Visual Basic version 6.0) ในการพัฒนาและอ้างอิงการทำงานทั้งในส่วนการเชื่อมโยงกับฐานข้อมูล รวมทั้งการรับข้อมูล (Input) และการแสดงผลออกทางหน้าจอ (Output) การพัฒนาโปรแกรมนี้ทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 95 ขึ้นไป

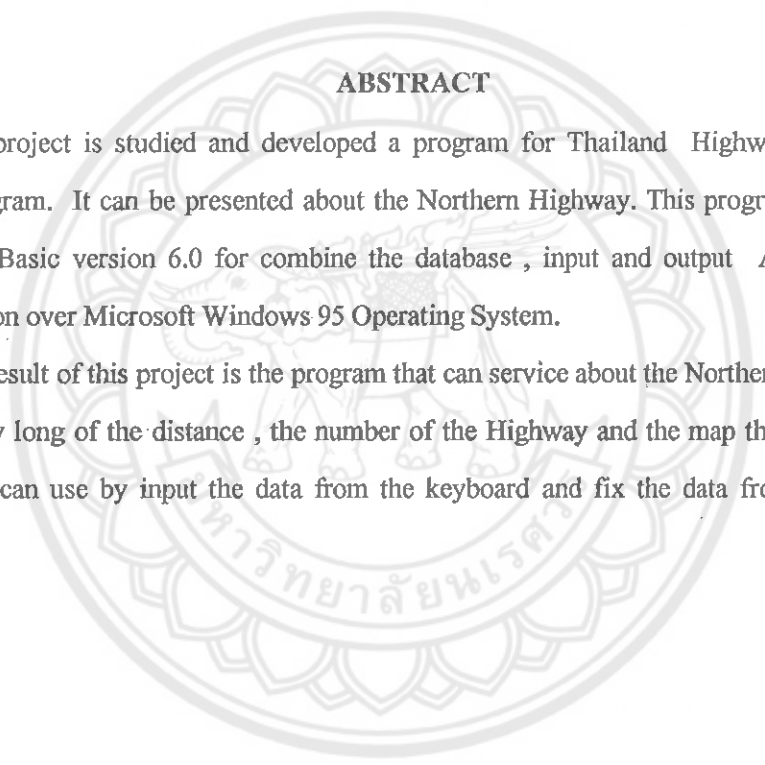
ผลที่คาดว่าจะได้รับจากการทำโครงการนี้ คือ ได้โปรแกรมที่สามารถให้บริการ เกี่ยวกับการให้ข้อมูลแผนที่ทางหลวงแผ่นดิน ในเขตภาคเหนือล่าง ซึ่งจะแสดงระยะทาง และหมายเลขทางหลวงที่ใช้ในการเดินทาง รวมทั้งแสดงแผนที่ของเส้นทางด้วย ซึ่งจะรับข้อมูลเข้ามา 2 ทาง คือ การกำหนดเส้นทางและปลายทางจากการพิมพ์ข้อมูลเข้าไป และการกำหนดเส้นทางและปลายทางจากแผนที่

Project Title	Electronic Map
Name	Mr.Chatchai Jaikemkhaeng ID.41360256
	<del>Mr.Wuttiptime Arkanit ID.41360439</del>
Project Advisor	Mr.Sithichok Chouwakul
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Year	2001

**ABSTRACT**

This project is studied and developed a program for Thailand Highway , by manage to Absolute Program. It can be presented about the Northern Highway. This program is developed by using Visual Basic version 6.0 for combine the database , input and output A development this program runs on over Microsoft Windows 95 Operating System.

The result of this project is the program that can service about the Northern Highway , which is present how long of the distance , the number of the Highway and the map that the user request. The program can use by input the data from the keyboard and fix the data from the map that is presented.



## กิตติกรรมประกาศ

การจัดทำโครงการวิศวกรรมศาสตร์ครั้งนี้สำเร็จลุล่วงด้วยดี เนื่องมาจากการแนะนำและความช่วยเหลือจากอาจารย์สิทธิโชค เขาวกุล และอาจารย์ประทีป ศิริธณ โสภาส ที่ได้ให้คำปรึกษาชี้แนวทางที่เป็นประโยชน์อย่างสูงในการทำโครงการครั้งนี้และขอขอบคุณเพื่อนๆ สำหรับความช่วยเหลือในด้านต่างๆ

นายฉัตรชัย ใจเข้มแข็ง  
นายวุฒิภูมิ อกนิษฐ์



# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	ก
บทคัดย่อภาษาอังกฤษ .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ .....	ง
สารบัญตาราง .....	ฉ
สารบัญรูป .....	ช

## บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ .....	1
1.3 ขอบข่ายของโครงการ .....	1
1.4 ขั้นตอนการดำเนินโครงการ .....	1
1.5 แผนการดำเนินงานตลอดโครงการ .....	2
1.6 รายละเอียดงบประมาณตลอดโครงการ .....	3

## บทที่ 2 โครงสร้างข้อมูล อัลกอริทึมและรูปแบบของโปรแกรม

2.1 ประเภทของถนนตามหลักกรมทางหลวง .....	4
2.2 การจัดเก็บข้อมูลเส้นทาง .....	6
2.3 อัลกอริทึมการค้นหาเส้นทางที่สั้นที่สุด .....	8
2.4 รูปแบบของโปรแกรม .....	10

## บทที่ 3 วิธีการดำเนินงาน

3.1 ออกแบบและพัฒนาโปรแกรม .....	18
---------------------------------	----

## บทที่ 4 การทดสอบและวิเคราะห์โปรแกรม

4.1 ลักษณะทั่วไปของโปรแกรม .....	26
4.2 วิธีการใช้งานของโปรแกรม .....	26
4.3 วิธีการทดสอบ .....	27

บทที่ 5 บทสรุป	
5.1 สรุปผล .....	30
5.2 ปัญหาและวิธีการในการแก้ปัญหา .....	30
5.3 แนวทางในการพัฒนาโปรแกรมต่อไป .....	30
ภาคผนวก ก .....	31
ภาคผนวก ข .....	34
เอกสารอ้างอิง .....	60
ประวัติผู้เขียน .....	61



## สารบัญตาราง

ตารางที่	หน้า
1.1 ผังการปฏิบัติงานปีการศึกษา 2/2543 .....	2
1.2 ผังการปฏิบัติงานปีการศึกษา 2544 .....	2
2.1 มาตรฐานชั้นทางสำหรับทางหลวงทั่วไป .....	5
2.2 ความสัมพันธ์ and .....	12
2.3 ความสัมพันธ์ or .....	12
2.4 แสดงความสัมพันธ์ not .....	12
2.5 ความสัมพันธ์ xor .....	12
2.6 ความสัมพันธ์ imp .....	13
2.7 ความสัมพันธ์ eqv .....	13
2.8 การประกาศฟังก์ชัน .....	15
2.9 การประกาศโปรแกรมย่อย .....	16
2.10 คำสั่ง call .....	16
2.11 วิธีเรียก โพรซีเยอร์ .....	16
3.1 ค่าพรีอพเพอร์ตี้ของฟอร์มที่ 1 .....	18
3.2 ค่าพรีอพเพอร์ตี้ของฟอร์มที่ 2 .....	18



# สารบัญรูป

รูปที่	หน้า
2.1 Tree .....	6
2.2 Graph .....	7
2.3 กราฟของโหนด x y .....	7
2.4 การเชื่อมโยง Node .....	9
2.5 พิกัดของจอภาพและฟอร์ม .....	14
2.6 Client Area .....	15
3.1 ฟอร์มที่ 1 .....	19
3.2 ฟอร์มที่ 2 .....	20
3.3 ส่วนประกอบต่างๆของฟอร์มข้อมูล .....	21
3.4 ผลลัพธ์เมื่อทำการ Run โปรแกรมจากการจัดเก็บข้อมูล .....	22
3.5 แผนผังงาน โดยรวมของโปรแกรม .....	23
4.1 ผลจากการทดสอบในส่วนของการกดเมาส์ .....	27
4.2 ผลจากการทดสอบในส่วนของการรับข้อมูลผ่านทางกรอกข้อมูล.....	28
ก.1 การติดตั้งโปรแกรมแสดงแผนที่ .....	32
ก.2 การเลือก Directory .....	32
ก.3 การเลือก Program Group .....	33

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากการจราจรมีความสำคัญในการดำรงชีวิตเป็นอย่างมาก ไม่ว่าจะเป็นการจราจรทางบก ทางเรือ และทางอากาศ แต่การจราจรที่ได้รับความนิยมมากที่สุดก็คือการจราจรทางบก ซึ่งอาจจะมีผลเนื่องมาจากมีความสะดวกสบาย รวดเร็ว และอื่นๆอีกมากมาย แต่หากผู้ใช้เส้นทางนั้นไม่ทราบเส้นทางที่จะไปสู่จุดหมายซึ่งเป็นเส้นทางที่ถูกต้อง หรือไม่ทราบว่าควรจะไปเส้นทางใดที่ใกล้จุดหมายปลายทางมากที่สุด การจราจรทางบกนั้นก็เลยไม่อำนวยความสะดวกแก่ผู้ใช้ได้ทันที

ดังนั้นทางคณะผู้จัดทำจึงมีแนวความคิดที่จะออกแบบ โปรแกรมแสดงแผนที่ (Electronic Map) เพื่อที่จะอำนวยความสะดวกแก่ผู้ใช้เส้นทางจราจรทางบก ให้ได้ใช้เส้นทางจราจรทางบกเพื่อเดินทางไปสู่จุดหมายปลายทางได้อย่างถูกต้อง และเพื่อเป็นต้นแบบในการศึกษาเพื่อนำไปประยุกต์ใช้ในด้านต่างๆสำหรับผู้สนใจต่อไป

#### 1.2 วัตถุประสงค์

- 1.2.1 เพื่อนำความรู้ที่ได้ศึกษามาทำประโยชน์ต่อสังคม
- 1.2.2 พัฒนารูปแบบการเดินทาง โดยใช้แผนที่ให้ง่ายและสะดวกมากขึ้น

#### 1.3 ขอบข่ายของโครงการ

- 1.3.1 ศึกษาโปรแกรม Visual Basic 6.0
- 1.3.2 ศึกษาโปรแกรมการหาระยะทางที่สั้นที่สุด (Shortest Path)
- 1.3.3 ศึกษาเส้นทางหลวงแผ่นดิน
- 1.3.4 ศึกษาโปรแกรม Photoshop

#### 1.4 ขั้นตอนการดำเนินงาน

- 1.4.1 ศึกษาโปรแกรม Visual Basic 6.0
- 1.4.2 ศึกษาโปรแกรมการหาระยะทางที่สั้นที่สุด (Shortest Path)
- 1.4.3 ศึกษาเส้นทาง หลวงแผ่นดิน
- 1.4.4 ศึกษาโปรแกรม Photoshop
- 1.4.5 ออกแบบและพัฒนาผลลัพธ์ที่ได้ออกทางหน้าจอ

1.4.6 ออกแบบและพัฒนาการจัดเก็บฐานข้อมูลเส้นทาง

1.4.7 ออกแบบและพัฒนาโปรแกรม ในการค้นหาเส้นทางที่ใช้เวลาน้อยที่สุด

1.4.8 พัฒนาโปรแกรม Visual Basic 6.0 ในส่วนหน้าจอ

1.4.9 พัฒนาโปรแกรม Visual Basic 6.0 ในการค้นหาเส้นทางที่ใช้เวลาน้อยที่สุด

1.4.10 ทดลองใช้งานจริง

1.4.11 ปรับปรุงและแก้ไข

1.4.12 สรุปผล

## 1.5 แผนการดำเนินงานตลอดโครงการ

ตารางที่ 1.1 ผังการปฏิบัติงานปีการศึกษา 2/2543

กิจกรรม	เดือน-ปี			
	พ.ย.2543	ธ.ค.2543	ม.ค.2544	ก.พ.2544
1.ศึกษาโปรแกรม Visual Basic 6.0	←→			
2.ศึกษาโปรแกรม Shortest Path	←→			
3. ศึกษาเส้นทาง หลวง แผ่นดิน		←→		
4. ศึกษาโปรแกรม Photoshop			←→	
5.สรุปผล				←→

ตารางที่ 1.2 ผังการปฏิบัติงานปีการศึกษา 2544

กิจกรรม	เดือน-ปี						
	มี.ค.44	เม.ย.44	พ.ค.44	มิ.ย.44	ก.ค.44	ส.ค.44	ก.ย.44
1.ออกแบบ หน้าจอ	←→						
2.ออกแบบ การจัดเก็บ ฐานข้อมูล	←→						

3. ออกแบบ และพัฒนา โปรแกรม ในการค้น							
หาเส้นทาง ที่ใช้เวลา น้อยที่สุด							
4. พัฒนา โปรแกรม Visual Basic 6.0 ในส่วนหน้า จอ							
5. พัฒนา โปรแกรม Visual Basic 6.0 ในการค้น หาเส้นทาง ที่ใช้เวลา น้อยที่สุด							
6. ทดลองใช้ งานจริง							
7. ปรับปรุง และแก้ไข							
8. สรุปผล							

ตารางที่ 1.2 (ต่อ)

### 1.6 รายละเอียดงบประมาณตลอดโครงการ

หนังสือ เขียนเกมต์จาก DirectX	จำนวน 1 เล่ม	เป็นเงิน 375 บาท
หนังสือ Visual Basic	จำนวน 1 เล่ม	เป็นเงิน 333 บาท
หมึก Printer Sepom	จำนวน 2 คลับ	เป็นเงิน 1030 บาท
กระดาษขนาด A4	จำนวน 2 รีม	เป็นเงิน 300 บาท
รวมเป็นเงิน		2038 บาท
		เบิกได้เป็นเงิน 2000 บาท

## บทที่ 2

# โครงสร้างข้อมูล อัลกอริทึมและรูปแบบของโปรแกรม

### 2.1 ประเภทของถนนตามหลักกรมทางหลวง

การกำหนดประเภทของถนนตามหลักกรมทางหลวงโดยกำหนดตามจำนวนหมายเลขของถนน ซึ่งจะกำหนดประเภทของถนนได้ดังนี้

1. ถนนที่มีหมายเลขเส้นทางเป็นตัวเลข 1 หลัก จะมีความหมาย คือ เป็นเส้นทางเดินถนนระหว่างภาคเชื่อมโยงกัน โดยมีรายละเอียดดังนี้

- 1.1 ภาคเหนือ และภาคกลาง จะมีการเชื่อมต่อกันโดยใช้หมายเลขเส้นทางหลวงหมายเลข 1 ทางหลวงเส้นนี้จะเดินทางจาก กรุงเทพมหานคร ไปสู่อำเภอแม่สายจังหวัดเชียงราย
- 1.2 ภาคตะวันออกเฉียงเหนือ และภาคกลาง จะมีการเชื่อมต่อกันโดยใช้หมายเลขเส้นทางหลวงหมายเลข 2 ทางหลวงเส้นนี้จะเดินทางจาก สระบุรี ไปสู่อุทยานมิตรภาพไทย-ลาว (จังหวัดหนองคาย)
- 1.3 ภาคตะวันออก และภาคกลาง จะมีการเชื่อมต่อกันโดยใช้หมายเลขเส้นทางหลวงหมายเลข 3 ทางหลวงเส้นนี้จะเดินทางจาก กรุงเทพมหานคร ไปสู่อำเภอตราด
- 1.4 ภาคใต้ และภาคกลาง จะมีการเชื่อมต่อกันโดยใช้หมายเลขเส้นทางหลวงหมายเลข 4 ทางหลวงเส้นนี้จะเดินทางจาก กรุงเทพมหานคร ไปสู่อำเภอสะเตาะ จังหวัดสงขลา

2. ถนนที่มีหมายเลขเส้นทางเป็นตัวเลข 2 หลัก จะมีความหมาย คือ เป็นเส้นทางเดินถนนระหว่างจังหวัด เชื่อมโยงกับถนนระหว่างภาค เช่น ทางหลวงหมายเลข 11 ทางหลวงเส้นนี้จะเดินทางจาก อำเภออินทร์บุรี จังหวัดสิงห์บุรี ไปสู่อำเภอเมือง จังหวัดเชียงใหม่ ทางหลวงหมายเลข 12 ทางหลวงเส้นนี้จะเดินทางจาก อำเภอขอนแก่น ไปสู่อำเภอตาก

3. ถนนที่มีหมายเลขเส้นทางเป็นตัวเลข 3 หลัก จะมีความหมาย คือ เป็นเส้นทางเดินถนนระหว่างอำเภอ เชื่อมโยงกับถนนระหว่างจังหวัด เช่น ทางหลวงหมายเลข 117 ทางหลวงเส้นนี้จะเดินทางจาก อำเภอพิบูลย์รักษ์ ไปสู่อำเภอรัตนบุรี จังหวัดบุรีรัมย์ ทางหลวงหมายเลข 115 ทางหลวงเส้นนี้จะเดินทางจาก อำเภอพิบูลย์รักษ์ ไปสู่อำเภอแก่งเพชร

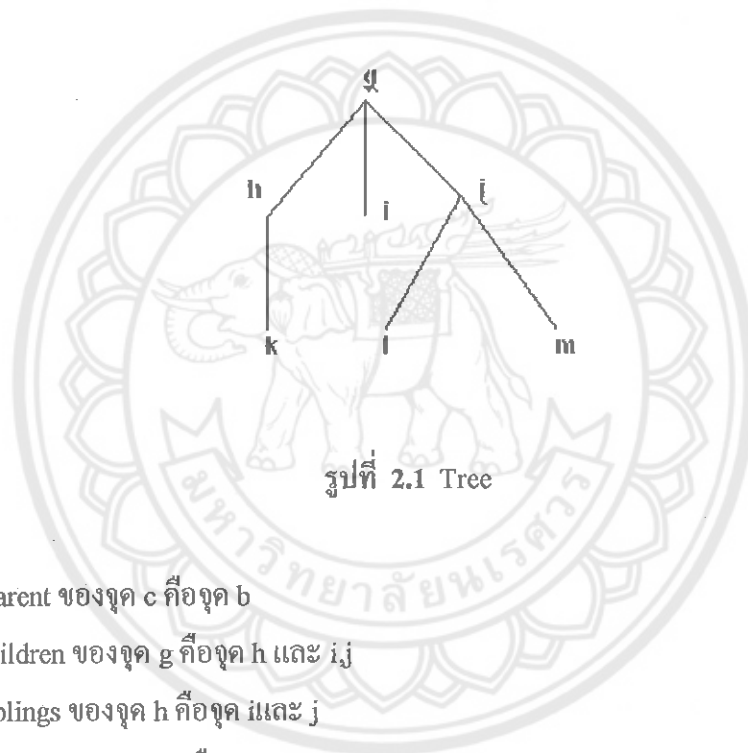
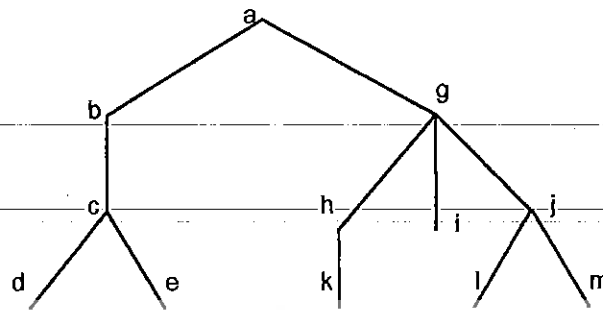
4. ถนนที่มีหมายเลขเส้นทางเป็นตัวเลข 4 หลัก จะมีความหมาย คือ เป็นเส้นทางเดินถนนระหว่างอำเภอ เช่น ทางหลวงหมายเลข 1313 ทางหลวงเส้นนี้จะเดินทางจาก อำเภอตะพานหิน ไปสู่อำเภอบางมูลนาก ทางหลวงหมายเลข 1069 ทางหลวงเส้นนี้จะเดินทางจาก อำเภอบางมูลนาก ไปสู่อำเภอคงเจริญ

ตารางที่ 2.1 มาตรฐานชั้นทางสำหรับทางหลวงทั่วไป

ชั้นทาง	พิเศษ	1	2	3	4	เขตเมือง	ทางขนาน
ปริมาณการจราจรเฉลี่ยทั้งวัน	มากกว่า	4,000-	2,000-	1,000-	300-	-	-
	8,000	8,000	4,000	2,000	1,000		
อัตราความเร็วที่ใช้ออกแบบ กม./ชม.							
- ทางราบ		90-110			70-90	60	70-90
- ทางเนิน		80-110			55-70	60	70-80
- ทางเขา		70-90			40-55	60	60-70
ความลาดชันสูงสุด %							
- ทางราบ		4			4	ตามพื้นที่	4
- ทางเนิน		6			8	ตามพื้นที่	6
- ทางเขา		8			12	ตามพื้นที่	8
ประเภทผิวจราจรที่เสนอแนะ		ชั้นสูง		กลาง-สูง		ชั้นสูง	กลาง-สูง
ความกว้างของผิวจราจร (เมตร)	อย่างน้อย 7.00	7.00	6.50	6.00	6.00	ช่องทางจราจรละ 3.00-3.50	ช่องทางจราจรละ 3.00-3.50
ความกว้างของไหล่ทาง (เมตร)	ชาย 2.50-3.00 ขวา 1.50	2.50-3.00	2.25	2.00	1.50	2.50 หรือเป็นทางเข้า	อย่างน้อย 2.00 เมตร หรือเป็นทางเท้า
ความกว้างของผิวทางจราจรสะพาน (เมตร)							
1. สะพานทั่วไป	11.00 (MIN.)	12.00	11.00	10.00	9.00	สะพานกว้างเต็มคันทาง ขอบข้างทางละ 0.50 เมตร	
2. สะพานในย่านชุมชน	9.50 (MIN.)	9.00	9.00	8.00	8.00	ทางเท้าข้างละ 1.50 เมตร	
3. สะพานที่ยาวเกิน 80.00 (เมตร)							
3.1 มีทางเท้า	9.50	9.00	9.00	8.00	8.00	ทางเท้าข้างละ 1.50 เมตร	
3.2 ไม่มีทางเท้า	10.00	10.00	10.00	9.00	9.00	ขอบทางข้างละ 0.50 เมตร	
ความกว้างของเขตทาง (เมตร)		60-80	40-60	30-40		ตามความเหมาะสม	-
ยกโค้งราบสูงสุด		10%				6%	10%

## 2.2 การจัดเก็บข้อมูลเส้นทาง

### 2.2.1 ต้นไม้ Trees



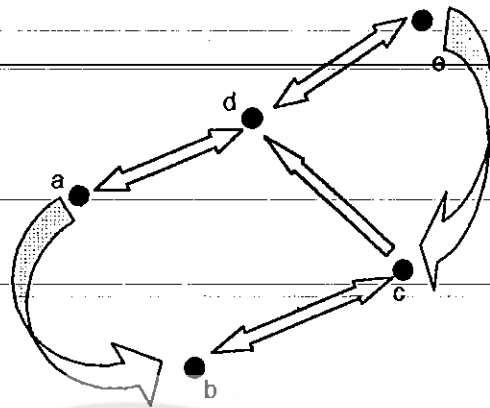
รูปที่ 2.1 Tree

- Parent ของจุด c คือจุด b
- children ของจุด g คือจุด h และ i,j
- siblings ของจุด h คือจุด i และ j
- ancestors ของจุด e คือจุด c,b และ a
- descendants ของจุด b คือจุด c,d และ e
- internal vertices คือจุด a,b,c,g,h และ j
- leaves คือจุด d,e,f,i,k,l และ m
- subtree rooted ที่จุด g แสดงในรูปสี่เหลี่ยม

### 2.2.2 นิยามกราฟ (Graph Definition) ของ George F. Loger กล่าวว่า

- กราฟคือเซตของ Nodes ที่มี Arcs เชื่อมต่อระหว่างคู่ Nodes
- Direct Graph คือกราฟที่มีทิศทางแสดง โดยใช้หัวลูกศร
- Path คือลำดับทางเดินของ Nodes ที่มี Arcs เชื่อมถึงกัน [a,b,c,d]
- Loops หรือ Cycle คือ Path ที่ปรากฏ Nodes มากกว่าหนึ่งครั้ง

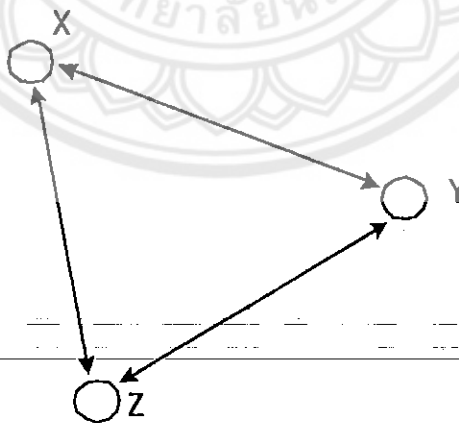
- Tree คือ Direct Graph ที่มี Loop



รูปที่ 2.2 Graph

คำจำกัดความของ สำรวม จงเจริญ

กราฟ  $G = (V, E)$  ประกอบด้วยเซตของโหนด (Vertices)  $V$  และเซตของกิ่ง (edges)  $E$  กิ่งในบางครั้งเรียกว่า ขอบโค้ง (arcs) ในกราฟไม่ระบุทิศทาง (undirected graph) แต่ละกิ่ง  $e = \{v, w\}$  เป็นคู่อันดับของ  $(v, w)$  โดยที่  $v, w \in V$  ในกราฟระบุทิศทาง (directed graph) แต่ละกิ่ง  $e = (v, w)$  เป็นคู่อันดับที่มีอันดับ (order pair) หมายถึงเขียนแทนด้วยลูกศรจาก  $v$  ไป  $w$  หรือ  $w$  ไป  $v$  กราฟระบุทิศทางบางครั้งเรียกว่าไดกราฟ (digraph) โหนด  $w$  เชื่อมต่อ (adjacent) โหนด  $v$  ถ้า  $v, w \in V$



รูปที่ 2.3 กราฟของโหนด x y

ทางเดินจากโหนด  $x$  ไป  $y$  ในกราฟคือรายการของโหนดซึ่งโหนดทั้งหลายที่ตามมาต้องถึงกันด้วยกิ่งในกราฟ กราฟเชื่อมต่อ (connected graph) ถ้ามันมีทางเดินจากทุก ๆ โหนดไปยังโหนดอื่นใน



กราฟ ทางเดินอย่างง่าย (simple path) คือทางเดินที่ไม่ซ้ำโหนดเดิม ทางเดินวัฏจักร (cycle) คือทางเดินอย่างง่ายที่โหนดแรกและโหนดสุดท้ายของทางเดินเป็นโหนดเดียวกัน

กราฟที่ไม่มีทางเดินวัฏจักรเรียกว่าต้นไม้ กลุ่มของต้นไม้ที่ไม่ต่อดังกันเรียกว่าป่า ต้นไม้ของกราฟ (spanning tree) คือกราฟย่อยที่บรรจุโหนดทั้งหมดแต่มีกิ่งเท่าที่เพียงพอต่อเป็นต้นไม้

เมื่อเราพูดถึงกราฟโดยไม่มีคำขยาย ปกติเรามักจะหมายถึงกราฟไม่ระบุทิศทาง และเราจะใช้คำว่า digraph และ directed graph สำหรับกราฟที่ถึงขอบมันระบุทิศทาง

ในกราฟระบุน้ำหนัก (weighted graph) เลขจำนวนเต็มหรือน้ำหนักถูกกำหนดให้แต่ละโหนดเพื่อแทนเช่น ระยะทาง หรือต้นทุน เป็นต้น กราฟระบุทิศทางและน้ำหนักบางครั้งเรียกว่าเครือข่าย (networks)

### การแทนกราฟ

ในการประมวลผลกราฟด้วยโปรแกรมคอมพิวเตอร์ ก่อนอื่นเราต้องตัดสินใจว่าจะแทนมันภายในคอมพิวเตอร์อย่างไร เรามองสองวิธีที่นิยมใช้กันในการแทน การเลือกว่าจะแทนด้วยวิธีใดขึ้นกับว่ากราฟนั้นหนาแน่น (dense) หรือ โหล่ง (sparse) แต่อย่างไรก็ตามลักษณะของการกระทำที่กระทำก็มีบทบาทที่สำคัญ

ขั้นตอนแรกในการแทนกราฟคือ การแปลงชื่อโหนดเป็นจำนวนเต็มระหว่าง 1 ถึง  $V$  เหตุผลหลักสำหรับการทำเช่นนี้คือทำให้เข้าถึงสารสนเทศที่สอดคล้องกับแต่ละโหนดได้รวดเร็ว โดยใช้ดัชนีอาเรย์ รูปแผนการค้นหาใด ๆ ที่มาตรฐานสามารถใช้กับจุดประสงค์นี้ ตัวอย่างเราสามารถแปลงชื่อโหนดทั้งหมดเป็นเลขจำนวนเต็มระหว่าง 1 ถึง  $V$  และเก็บชื่อเหล่านั้นไว้ในตารางแฮชหรือต้นไม้ไบนารีที่สามารถค้นหาเพื่อพบจำนวนเต็มที่สอดคล้องกับชื่อโหนดใด ๆ เนื่องจากเราได้ศึกษาเทคนิคต่างๆ เหล่านี้ได้เรียบร้อยแล้ว สมมติว่าเรามีฟังก์ชัน index ในการแปลงจากชื่อโหนดเป็นเลขจำนวนเต็มระหว่าง 1 ถึง  $V$  และฟังก์ชัน name ในการแปลงจากเลขจำนวนเต็มเป็นชื่อ เพื่อให้อัลกอริทึมของเราง่ายในการศึกษา เราใช้หนึ่งตัวอักษรสำหรับชื่อโหนด โดยที่อักษรตัวที่  $I$  ของตัวอักษรสอดคล้องกับเลขจำนวนเต็ม  $I$  ถึงแม้ว่า name และ index จะเป็นเรื่องเล็กน้อยในการดำเนินการสำหรับตัวอย่างของเรา การใช้มันทำให้ง่ายในการขยายอัลกอริทึมเพื่อจัดการกับกราฟที่มีชื่อ โหนดจริง

## 2.3 อัลกอริทึมการค้นหาเส้นทางที่สั้นที่สุด (Shortest Path)

Network traversals การเชื่อมโยงของโครงข่าย Network ในส่วนของโปรแกรมจะใช้รูปแบบการเก็บข้อมูลไว้ใน Tree และจะใช้ทฤษฎีการค้นหาข้อมูลเพื่อหาเส้นทางที่สั้นที่สุด เพื่อช่วยต่อการค้นหาข้อมูลที่เก็บการเชื่อมต่อเส้นทางแต่ละเส้นทางไว้ โดยการค้นหานั้นมีหลายรูปแบบ โดยได้มีการจัดแบ่งรูปแบบการค้นหาตามหัวข้อดังนี้

- Un – Informed (Blind Search) เป็นการค้นหาข้อมูล ที่ทำการค้นหาข้อมูลทั้งหมดในกราฟ
- Breadth – First Search เป็นการค้นหาข้อมูลตามแนวกว้าง

- Depth – First Search เป็นการค้นหาข้อมูลตามแนวลึก
- Informed (Heuristic Search) วิธีการ Search ที่มีการใช้ข้อมูล โดยจะค้นหาบางส่วนของข้อมูล
- Best – First Search จะไม่ค้นหาทั้งกราฟ โดยจะค้นหาแค่เพียงแนวทางที่เป็นไปได้สู่จุดหมาย

ปลายทางที่ต้องการ

- Hill – Climbling Search
- Algorithm A\*

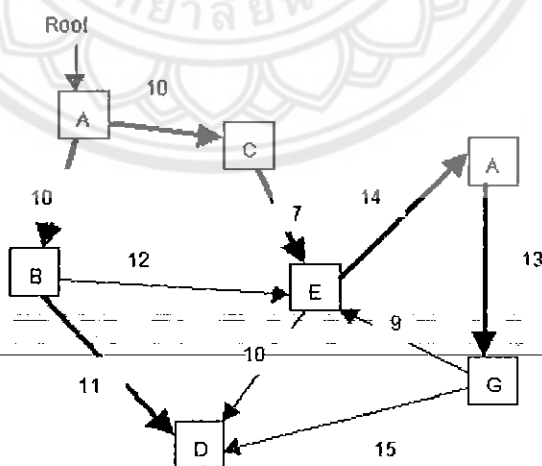
แต่ในการอธิบายในครั้งนี้จะนำเสนอเพียงการค้นหาข้อมูลสองรูปแบบ คือ Breadth First Search

และ Depth First Search

- Breadth-First Search การค้นหาตามแนวกว้างก่อน

การค้นหาตามแนวกว้างก่อนสอดคล้องกับการเคลื่อนตามต้นไม้ (tree) เป็นลำดับ (level) โดยเคลื่อนที่ไปยังโหนดใกล้เคียง ไปทางด้านข้างของโหนดที่กำหนดไว้ จากนั้นก็ไปยังโหนดใกล้เคียงในลำดับถัดไป หมายความว่า เราจะค้นหาไปตามแนวกว้างในการค้นหาก่อนที่จะลงลึกในลำดับต่อไป หรือกล่าวอีกอย่างหนึ่งว่า การค้นหาตามแนวกว้างก่อนหรือการค้นหาตามแนวระดับเป็นการค้นหาในรูปแบบลักษณะคล้ายเครือข่าย Star คือมีจุดศูนย์กลางคือโหนดแรกแล้วแยกออกไปจนครบทุกกิ่ง จากนั้นก็เปลี่ยนศูนย์กลางไปยังโหนดระดับต่อไปที่ใกล้เคียงกับโหนดแรก แล้วแต่ละโหนดจะแยกออกไปทำดังกล่าวเรื่อยๆ ทีละระดับ

Depth Frist Search เป็นการค้นหาข้อมูลในแนวแกนลึกก่อนเป็นกรณีทั่วไปของ preorder โดยจะมีการค้นหาจาก Root ไปยังโหนดทุกโหนด ของ Tree และตรวจสอบทุกกิ่งในกราฟอย่างเป็นระเบียบ



รูปที่ 2.4 การเชื่อมโยง Node

รูปแบบโครงสร้างของ Shortest Path สามารถอธิบายได้โดยการค้นหา เส้นทางที่สั้นที่สุดจาก Root ไปยัง โหนดต่างๆใน Network โดยรูปแบบการเชื่อมโยงจะเป็นแบบ Tree ดังรูปที่ 2.4

จากรูปจะแสดงการเชื่อมโยงเพื่อหาเส้นทางที่สั้นที่สุดโดยจะเริ่มจาก Root A โดย Tree จะแสดงเส้นทางที่สั้นที่สุด จาก Node A ไปยัง ทุกๆ Node ใน Network สิ่งที่สำคัญที่สุดของโครงสร้างในส่วน ของ Program นี้ก็คือ การกำหนดจุดเริ่มต้น(Start) จุดสุดท้าย(Empty) และการเพิ่ม Link เข้ากับการเชื่อม ต่อจาก Node แต่ละ Node ของ Tree จนกระทั่ง Tree สมบูรณ์โดยโครงสร้างของ Program จะเป็นแบบ Label Correcting Algorithms เป็นการเพิ่ม Link ซึ่งอาจจะเป็นเส้นทางที่สั้นที่สุดหรือไม่ก็ได้ โดย Algorithm นี้จะไม่ครอบคลุม Link ต่าง ๆ ที่เข้ามาแทนที่ในส่วนของ Tree แต่จะเก็บ Link เก่าและ Link ต่อ ๆ ไป โดยการเก็บ Link ใน Tree อาจจะทำเส้นทางใหม่ ซึ่งเป็นไปได้ที่ไม่เหมือนเส้นทางเดิม ในการตรวจสอบเส้นทางจะหาเส้นทางใหม่ก่อน

โดยในส่วนของ Program นี้จะใช้ในส่วนของ Label Correcting โดยมีขบวนการดังนี้

1. เริ่มต้นด้วยการ Set ค่าระยะทางที่จุดเริ่มต้น = 0
2. กำหนด Root Node ใน การเชื่อมโยง โดยจะเลือก Node แรก และจะเพิ่มต่อไปเรื่อย ๆ เพื่อหาเส้นทางที่สั้นที่สุด โดย Algorithm นี้จะตรวจสอบ Node ใกล้เคียงเพื่อเลือก Node และ Node ที่ถูกเลือก จะเชื่อมโยงโดย Link
3. ตรวจสอบระยะทางจาก Root ไปยัง Node ที่ถูกเลือก และรวมระยะทางจาก Root ไปถึง Node ที่ถูกเลือก ถ้ามีค่าน้อยกว่าเส้นทางเดิม ให้เลือกเส้นทางนี้เป็นเส้นทางที่ดีที่สุด Algorithm นี้จะถูก เปลี่ยนแปลงไปเรื่อยๆ ( Up Date ) จาก Node ไปยัง Node ใกล้เคียงเพื่อหาเส้นทางที่สั้นที่สุด โดย Algorithm นี้จะ ไม่นำค่าก่อนหน้ามาคิด
4. Algorithm นี้จะเปลี่ยนแปลง Node และตรวจสอบ Node ใกล้เคียงจนกระทั่งถึง Node สุดท้าย Label Correcting จะเลือก Node ระยะทาง แรกจากรายการ ซึ่งมันอาจจะไม่เป็นสิ่งที่ดีที่สุด ก็ได้โดย มันจะเปลี่ยนแปลง ไปเรื่อยๆ จนจบ และเลือกสิ่งที่ดีที่สุด

## 2.4 ส่วนของโปรแกรม

### 2.4.1 ศึกษาโปรแกรม Visual Basic

ศึกษาประโยคคำสั่งที่ต้องใช้ในการพัฒนา โปรแกรมแสดงแผนที่ โดยสามารถแบ่งออกเป็นสาม ประเภทหลักๆดังนี้

1. ประโยคกำหนดค่า (Assignment Statement) คือการเขียนคำสั่งที่ป้อนค่าหรือใส่ค่าให้กับตัวแปร ด้วยนิพจน์ต่างๆ ตัวดำเนินการหลักที่ใช้ใน ประโยคกำหนดค่าของ Visual Basic ก็คือเครื่องหมาย =
2. ประโยคแบบเงื่อนไข (Condition Statement) เป็นประโยคคำสั่งที่ใช้ในการตรวจสอบค่าตัดสินใจเปรียบเทียบ และเลือกประมวลผลส่วนของโปรแกรมที่เราต้องการ คำสั่งประเภทนี้ได้แก่ คำสั่ง IF-THEN, IF-THEN-ELSE, IF-THEN-ELSEIF และ SELECT-CASE

3. ประโยคแบบวนซ้ำ (Iteration Statement) เป็นประโยคคำสั่งที่ทำให้คอมพิวเตอร์ทำส่วนของโปรแกรมซ้ำๆกันหลายหน คำสั่งประเภทนี้ได้แก่ คำสั่ง FOR-NEXT, DO-WHILE LOOP, DO-UNTIL LOOP

ศึกษาชนิดของตัวแปรที่ต้องใช้ในการพัฒนาโปรแกรมแสดงแผนที่ซึ่งได้แก่ Integer, Single, Double, String โดยแต่ละชนิดมีหน้าที่ดังนี้

1. Integer มีหน้าที่เก็บเลขจำนวนเต็ม มีค่าตั้งแต่ -32,768 ถึง 32,767 ใช้หน่วยความจำเท่ากับ 2 ไบต์

2. Single มีหน้าที่เก็บเลขจำนวนเต็ม มีค่าตั้งแต่ -2,147,483,648 ถึง 2,147,483,647 ใช้หน่วยความจำเท่ากับ 4 ไบต์

3. Double มีหน้าที่เก็บเลขจำนวนจริงมีจุดทศนิยม โดยมีค่าตั้งแต่  $-1.797693134886232 \times 10^{308}$  ถึง  $-1.401298 \times 10^{-45}$  และค่าบวกตั้งแต่  $4.9406564588412471 \times 10^{-324}$  ถึง  $1.797693134886232 \times 10^{308}$  ใช้หน่วยความจำเท่ากับ 4 ไบต์

4. String มีหน้าที่เก็บกลุ่มของตัวอักษรหรือชุดของข้อความ (Character string) หน่วยความจำที่ใช้ขึ้นความยาวของข้อความ

5. Byte มีค่าตั้งแต่ 0 - 255 ใช้หน่วยความจำเท่ากับ 1 ไบต์

6. Currency ใช้เก็บเลขทศนิยมตำแหน่งจำกัด (Fixed Decimal Point) มีค่าตั้งแต่ 922,337,203,685,477.5808 ถึง -922,337,203,685,477.5808 ใช้หน่วยความจำเท่ากับ 8 ไบต์

7. Boolean ใช้เก็บค่าทางตรรกะ (logic) คือ True และ False ใช้หน่วยความจำเท่ากับ 2 ไบต์

8. Date ใช้เก็บตัวแปรที่เป็นเวลา หรือ วันที่ ใช้หน่วยความจำเท่ากับ 8 ไบต์

9. Object ใช้เก็บตัวแปรที่อ้างถึงออบเจ็คใดๆ ใช้หน่วยความจำเท่ากับ 4 ไบต์

10. Variant เป็นตัวแปรพิเศษของ Visual Basic สามารถใช้เก็บตัวแปรได้ทุกแบบข้างต้น

ศึกษาขอบเขตของตัวแปรที่ต้องใช้ในการพัฒนาโปรแกรมแสดงแผนที่ โดยแบ่งตามการใช้งานได้สามอย่างดังนี้

1. ตัวแปร Public มองเห็นและเรียกใช้ได้ในทุกฟอร์ม, โมดูล หรือทุกโปรซีเยอร์

2. ตัวแปร Private มองเห็นและเรียกใช้ได้เฉพาะทุกโปรซีเยอร์ใน โมดูลหรือฟอร์มที่ประกาศเท่านั้น

3. ตัวแปร Local มองเห็นและเรียกใช้ได้เฉพาะโปรซีเยอร์ที่ประกาศเท่านั้น

ศึกษาตัวดำเนินการที่ต้องใช้ในการพัฒนาโปรแกรมแสดงแผนที่ โดยจะมีการแบ่งตัวดำเนินการออกเป็น 3 กลุ่มคือตัวดำเนินการทางคณิตศาสตร์ (Arithmetic) ตัวดำเนินการเปรียบเทียบ (Relational) และตัวดำเนินการทางตรรกะ (Logical) ซึ่งมีรายละเอียดดังนี้

1. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator) เป็นตัวดำเนินการที่ใช้คำนวณทางคณิตศาสตร์ เช่น +, -, \*, /, และ ^ (ยกกำลัง) เป็นต้น

2. ตัวดำเนินการเปรียบเทียบ (Relation Operator) หรือเรียกอีกอย่างว่า Comparison Operator

เป็นตัวดำเนินการที่ใช้ประเมินค่าข้อมูล ผลลัพธ์ที่ได้จะมีค่าเป็นจริง (True) หรือเท็จ (False) เท่านั้น ตัวอย่างได้แก่ =, >, <, <>, >= และ <= เป็นต้น

3. ตัวดำเนินการทางตรรกะ (Logical Operator) จะเป็นตัวดำเนินการที่ให้ค่าเป็นจริง (True) หรือเท็จ (False) ตัวอย่างได้แก่ and, or, xor, not, eqv และ Imp เป็นต้น โดยจะมีค่าการกระทำดังนี้

ตารางที่ 2.2 and เทียบได้กับคำว่า “และ” จะมีค่าเป็นจริงก็ต่อเมื่อประโยคทั้งสองเป็นจริง ถ้ามีตรรกะต่างกัน and จะเป็นเท็จ

นิพจน์1	นิพจน์2	นิพจน์1 and นิพจน์2
0	0	0
0	1	0
1	0	0
1	1	1

\*1 = จริง (True), 0 = เท็จ (False)

ตารางที่ 2.3 or เทียบได้กับคำว่า “หรือ” จะมีค่าเป็นเท็จก็ต่อเมื่อประโยคทั้งสองเป็นเท็จ จะมีค่าเป็นเท็จก็เมื่อประโยคทั้งสองเป็นจริง และถ้ามีค่าตรรกะต่างกัน or ก็จะเป็นจริงด้วย

นิพจน์1	นิพจน์2	นิพจน์1 or นิพจน์2
0	0	0
0	1	1
1	0	1
1	1	1

\*1 = จริง (True), 0 = เท็จ (False)

ตารางที่ 2.4 not เทียบได้กับคำว่า “ไม่” เป็นการสลับค่าตรรกะของตัวแปรหรือนิพจน์ จากจริงเป็นเท็จหรือจากเท็จเป็นจริง

นิพจน์1	Not นิพจน์
0	1
1	0

\*1 = จริง (True), 0 = เท็จ (False)

ตารางที่ 2.5 xor หรือ Exclusive OR จะมีค่าเป็นจริงก็ต่อเมื่อประโยคทั้งสองมีค่าตรรกะต่างกัน ถ้ามีค่าตรรกะเหมือนกัน xor จะมีค่าเป็นเท็จ

นิพจน์1	นิพจน์2	นิพจน์1 xor นิพจน์2
0	0	0

ตารางที่ 2.5 (ต่อ)

0	1	1
1	0	1
1	1	0
*1 = จริง (True), 0 = เท็จ (False)		

ตารางที่ 2.6 imp มาจากคำว่า Implication หรือ ถ้า-แล้ว ในวิชาตรรกศาสตร์ ค่าผลลัพธ์จะเป็นเท็จก็ต่อเมื่อ นิพจน์ 1 จริง และนิพจน์ 2 เท็จเท่านั้น ในกรณีอื่นจะเป็นจริงหมด

นิพจน์ 1	นิพจน์ 2	นิพจน์ 1 and นิพจน์ 2
0	0	1
0	1	1
1	0	0
1	1	1
*1 = จริง (True), 0 = เท็จ (False)		

ตารางที่ 2.7 eqv มาจากคำว่า Equivalence หรือเทียบเท่า ค่าผลลัพธ์จะเป็นจริงก็ต่อเมื่อค่าตัวแปรหรือนิพจน์ที่นำมา Eqv กันมีค่าตรรกะเหมือนกัน

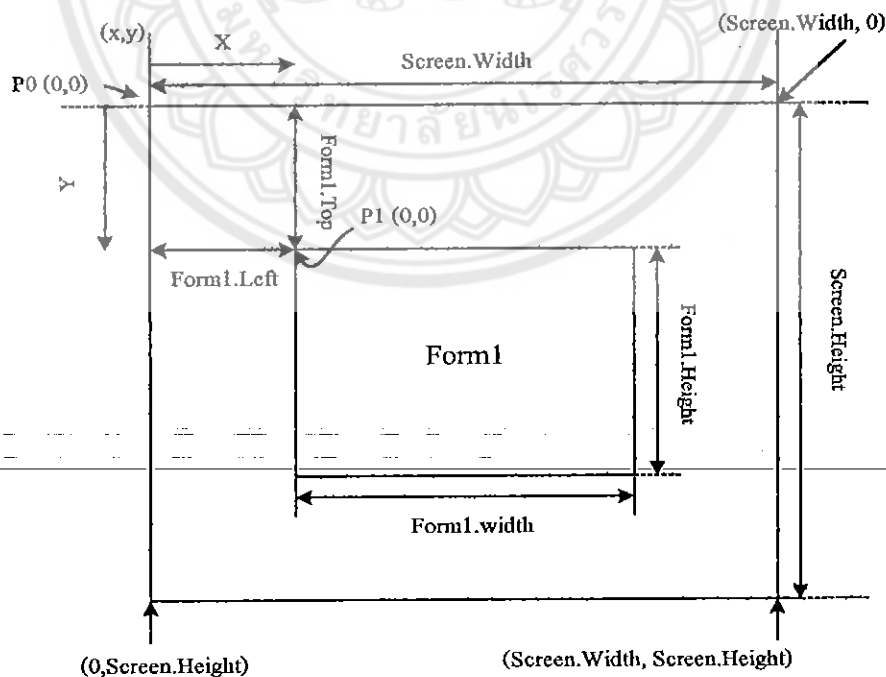
นิพจน์ 1	นิพจน์ 2	นิพจน์ 1 and นิพจน์ 2
0	0	1
0	1	0
1	0	0
1	1	1
*1 = จริง (True), 0 = เท็จ (False)		

ศึกษาคอนโทรลต่างๆที่จำเป็นต้องนำไปใช้ในการพัฒนาโปรแกรมแสดงแผนที่ ซึ่งมีดังนี้

1. Text Box
2. Command Button
3. Combo Box
4. Label
5. Frame
6. Image
7. Common Dialog และอื่นๆ

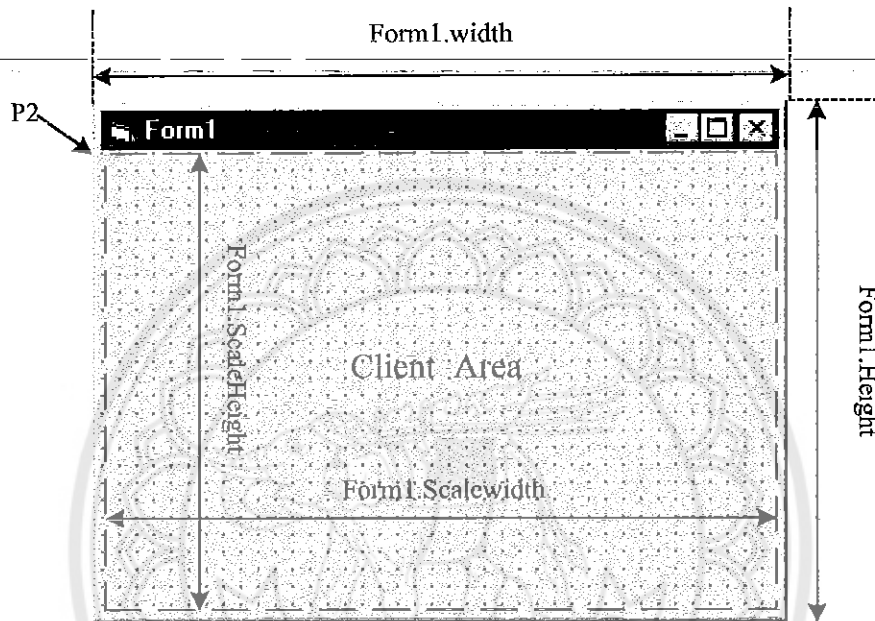
ศึกษาระบบพิกัดบนฟอร์ม (Coordinate System) โดยที่ให้หน่วยวัดขนาดและการกำหนดตำแหน่งบนฟอร์มจะมีค่าดีฟอลต์ ของหน่วยเป็น Twip โดย 1,440 Twip จะมีค่าเท่ากับ 1 นิ้ว สำหรับตำแหน่งพิกัดของจอภาพและฟอร์มใน Visual basic จะมีลักษณะดังนี้คือ

1. จุดใดๆของจอภาพสามารถบอกตำแหน่งพิกัดได้จากค่า (X,Y) ซึ่งจะเทียบจากตำแหน่งจุด P0 (0,0) ดังรูปที่ 2.5
2. X เป็นค่าพิกัดตำแหน่งในแนวนอน โดยซ้ายสุดมีค่าเป็น 0 และข้างขวาสุดจะมีค่ามากที่สุด สำหรับหน่วยที่ใช้จะเป็น Twip ที่ตั้งไว้ในพร็อพเพอร์ตี้ ScaleMode ดังรูปที่ 2.5
3. Y เป็นค่าพิกัดตำแหน่งในแนวตั้ง จุดบนสุดเป็น 0 และล่างสุดจะมีค่ามากที่สุด โดยหน่วยที่ใช้จะเป็น Twip ที่ตั้งไว้ในพร็อพเพอร์ตี้ ScaleMode ดังรูปที่ 2.5
4. เราสามารถทราบความกว้างและความสูงของจอภาพของระบบได้จากพร็อพเพอร์ตี้ Width และ Height ของ Screen (Screen.Width,Screen.Height) ดังรูปที่ 2.5
5. จุด P1 คือตำแหน่ง (0,0) จะเป็นตำแหน่งอ้างอิงของจุดทั้งหมดภายในฟอร์ม โดยจุดใดๆภายในฟอร์มสามารถบอกพิกัดตำแหน่งได้จากค่า (X,Y) ซึ่งจะเทียบจากตำแหน่งจุด P1 (0,0) ดังรูปที่ 2.5
6. X เป็นพิกัดในแนวนอน เทียบกับจุด P1 โดยซ้ายสุดเป็น 0 และทางขวาสุดจะมากที่สุด ดังรูปที่ 2.5
7. Y เป็นพิกัดในแนวตั้ง เทียบกับจุด P1 โดยจุดบนสุดเป็น 0 และล่างสุดจะมีค่ามากที่สุด ดังรูปที่ 2.5



รูปที่ 2.5 พิกัดของจอภาพและฟอร์ม

- 8. ตำแหน่งของฟอร์มบนหน้าจอภาพกำหนดได้จากพรีอเพอร์ตี Left และ Top ของฟอร์ม
- 9. ความกว้างและความสูงของฟอร์มกำหนดได้จากพรีอเพอร์ตี Width และ Height ของฟอร์ม
- 10. ความกว้างและความสูงของฟอร์มจากพรีอเพอร์ตี Width และ Height จะรวมเนื้อที่ของฟอร์มบางส่วนที่เราไม่สามารถใส่ข้อมูลได้ แต่เราจะสามารถทราบเนื้อที่เฉพาะที่เราใส่ข้อมูลได้ซึ่งเรียกว่า "Client Area" โดยหาขนาดได้จากพรีอเพอร์ตี ScaleHeight และ ScaleWidth ดังรูปที่ 2.6



รูปที่ 2.6 Client Area

- 11. ออบเจ็คใดๆที่อยู่บนฟอร์มจะมีตำแหน่ง Left, Top เทียบกับจุด P2 ซึ่งเป็นจุดเริ่มต้นของ ClientArea ที่อาจจะมีความเป็น (0,0) หรือ (Scaleleft, ScaleTop) ในกรณีกำหนด ScaleMode ไว้เป็น 0
- ศึกษาการประกาศฟังก์ชัน (Function) และ โปรแกรมย่อย (Sub Procedure) โดยมีรายละเอียดดังนี้
1. การประกาศฟังก์ชัน (Function) ก็คือการที่เราสามารถเขียนฟังก์ชันไว้สำหรับใช้เองได้ โดยมีรูปแบบดังตารางที่ 2.8

ตารางที่ 2.8 การประกาศฟังก์ชัน

รูปแบบ	Function ชื่อฟังก์ชัน (ตัวแปรพารามิเตอร์ As ชนิดตัวแปรที่รับเข้า) As ชนิดตัวแปรที่ส่งค่ากลับ
	ชุดคำสั่งต่างๆ
	ชื่อฟังก์ชัน = ...
	End Function



2. การประกาศโปรแกรมย่อย (Sub Procedure) เนื่องจากในการทำงานของโปรแกรมมักมีส่วนของโปรแกรมที่มีการทำงานซ้ำกันซึ่งถ้าซ้ำกันหลายทีก็สามารถแยกส่วนที่ซ้ำกันนั้นออกมาเพื่อประหยัดเนื้อที่ในหน่วยความจำ และความง่ายในการแก้ไขโปรแกรม การประกาศโปรแกรมย่อยนั้นทำได้โดยมีรูปแบบดังตารางที่ 2.9

ตารางที่ 2.9 การประกาศโปรแกรมย่อย

รูปแบบ	[Public/Private] Sub ชื่อโปรแกรมย่อย
	...
	[Exit Sub]
	End Sub

โดย Private และ Public จะเป็นการกำหนดขอบเขตของการเรียกใช้ Sub Procedure ที่เราสร้างขึ้น ถ้ากำหนดเป็น Private ทุกจุดในฟอร์มที่ประกาศเท่านั้นที่จะมองเห็น และเรียกใช้ Sub Procedure นี้ได้ แต่ถ้าประกาศเป็น Public นอกจากฟอร์มที่ประกาศแล้ว ฟอร์มอื่นใดที่อยู่ในโปรเจกต์จะสามารถมองเห็นและเรียกใช้ Sub Procedure ได้ด้วยเช่นกัน สำหรับการเรียกใช้สามารถทำได้โดยใช้คำสั่งดังตารางที่ 2.10

ตารางที่ 2.10 คำสั่ง Call

รูปแบบ	Call ชื่อโปรแกรมย่อย
--------	----------------------

หรือเรียกชื่อโปรแกรมย่อยโดยตรงดังตารางที่ 2.11

ตารางที่ 2.11 วิธีเรียกโปรแกรมย่อย

รูปแบบ	ชื่อโปรแกรมย่อย
--------	-----------------

ศึกษาตัวแปรแบบอาร์เรย์ (Array) และ Dynamic Array ซึ่งก็คืออาร์เรย์ปกติที่เราสามารถกำหนด Upper bound ของอาร์เรย์ได้ในภายหลังด้วยคำสั่ง Redim

ศึกษาฟังก์ชัน MsgBox และ InputBox

ศึกษาฟังก์ชัน String ซึ่งได้แก่ ฟังก์ชัน Right(), Left(), Mid() และ Len() เป็นต้น

ศึกษาการควบคุมเมาส์และการทำเมนู (Menu Editor)

ศึกษาชุดคำสั่งสำหรับจัดการข้อผิดพลาดในโปรแกรม ชุดคำสั่งนี้มี 3 คำสั่งคือ On Error Goto [Line], On Error Resume Next และ On Error Goto 0

ศึกษา Control Array คืออาร์เรย์ของคอนโทรลต่างๆที่มีชื่อ (Name) และชนิด (Type) เหมือนกัน แต่เราสามารถอ้างถึงคอนโทรลแต่ละตัวด้วยหมายเลขดัชนี (Index) ซึ่งมีได้ตั้งแต่ 0 ถึง 32,767 โดยแต่ละคอนโทรลนั้น สามารถมีพรีอเพอร์ตีของตนเองแตกต่างกันไปได้เสมือนคอนโทรลปกติทั่วไป

ศึกษาคำสั่งกำหนดสี ภาพ ฟังก์ชันเกี่ยวกับกราฟิก ซึ่งได้แก่ Line, Pset และ Circle เป็นต้น

#### 2.4.2 ศึกษาโปรแกรมหาเส้นทางที่สั้นที่สุด (Shortest Path)

โดยโปรแกรมที่ได้ศึกษานั้นจะหาเส้นทางที่สั้นที่สุด และเป็นการเดินทางแบบทางเดียว (One Way) ซึ่งมีวิธีการใช้โดยต้องใช้ Mouse คลิกที่จุดต้นทางและที่จุดปลายทาง ผลลัพธ์ที่ออกมาจะแสดงหมายเลขถนน และระยะทาง ของเส้นทางที่สั้นที่สุด

#### 2.4.3 ศึกษาโปรแกรม Adobe Photoshop 6.0

โดยได้ศึกษาคำสั่งที่ใช้ในการปรับความเข้มของแสง ความคมชัดของภาพ การตัดต่อภาพ และการขยายภาพ เป็นต้น

#### 2.4.4 ศึกษาเส้นทางหลวง

โดยได้ศึกษาหมายเลขเส้นทาง แบ่งได้เป็น ทางหลวงที่มีเลข 1 หลัก, 2 หลัก, 3 หลัก และ 4 หลัก เป็นต้น ได้ศึกษาระยะทางในแต่ละเส้นทาง โดยทั้งหมายเลขทางหลวงและระยะทางจะอ้างอิงมาจาก สำนักทางหลวงที่ 4 (พิษณุโลก)

## บทที่ 3

### วิธีการดำเนินงาน

#### 3.1 ออกแบบและพัฒนาโปรแกรม

##### 3.1.2 ออกแบบและพัฒนาหน้าจอ

หน้าจอที่ออกแบบและพัฒนาหน้านั้นจะมี 2 ฟอรัม ได้แก่ ฟอรัมที่เมื่อทำการรันโปรแกรมจะปรากฏขึ้นมา และฟอรัมที่เมื่อทำการกดปุ่มคำสั่งจะเข้ามาที่ฟอรัมนี้

ในฟอรัมที่ 1 จะประกอบไปด้วย Label ชื่อโปรแกรม, เมนู Help, Command Button สั่งให้ไปฟอรัมที่ 2 และ Command Button สั่งให้ออกจากโปรแกรม เป็นต้น ในฟอรัมที่ 1 นี้จะมีการกำหนดค่าพรีอเพอร์ติ์ที่สำคัญ ตามตารางที่ 3.1

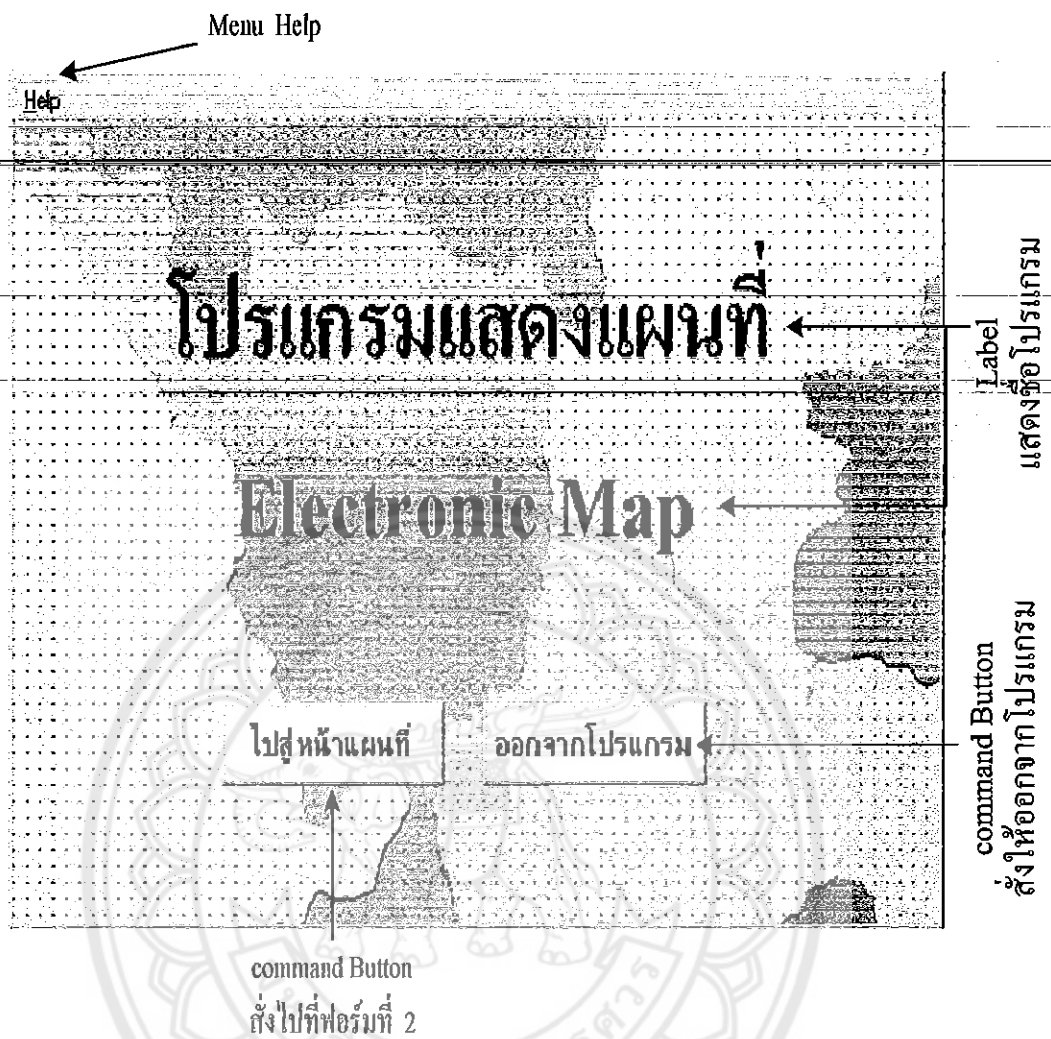
ตารางที่ 3.1 ค่าพรีอเพอร์ติ์ของฟอรัมที่ 1

Appearance	1-3D
BorderStyle	3-Fixed Dialog
Picture	Bitmap (กำหนดภาพเป็นพื้นหลัง)
StartPosition	2-Centerscreen

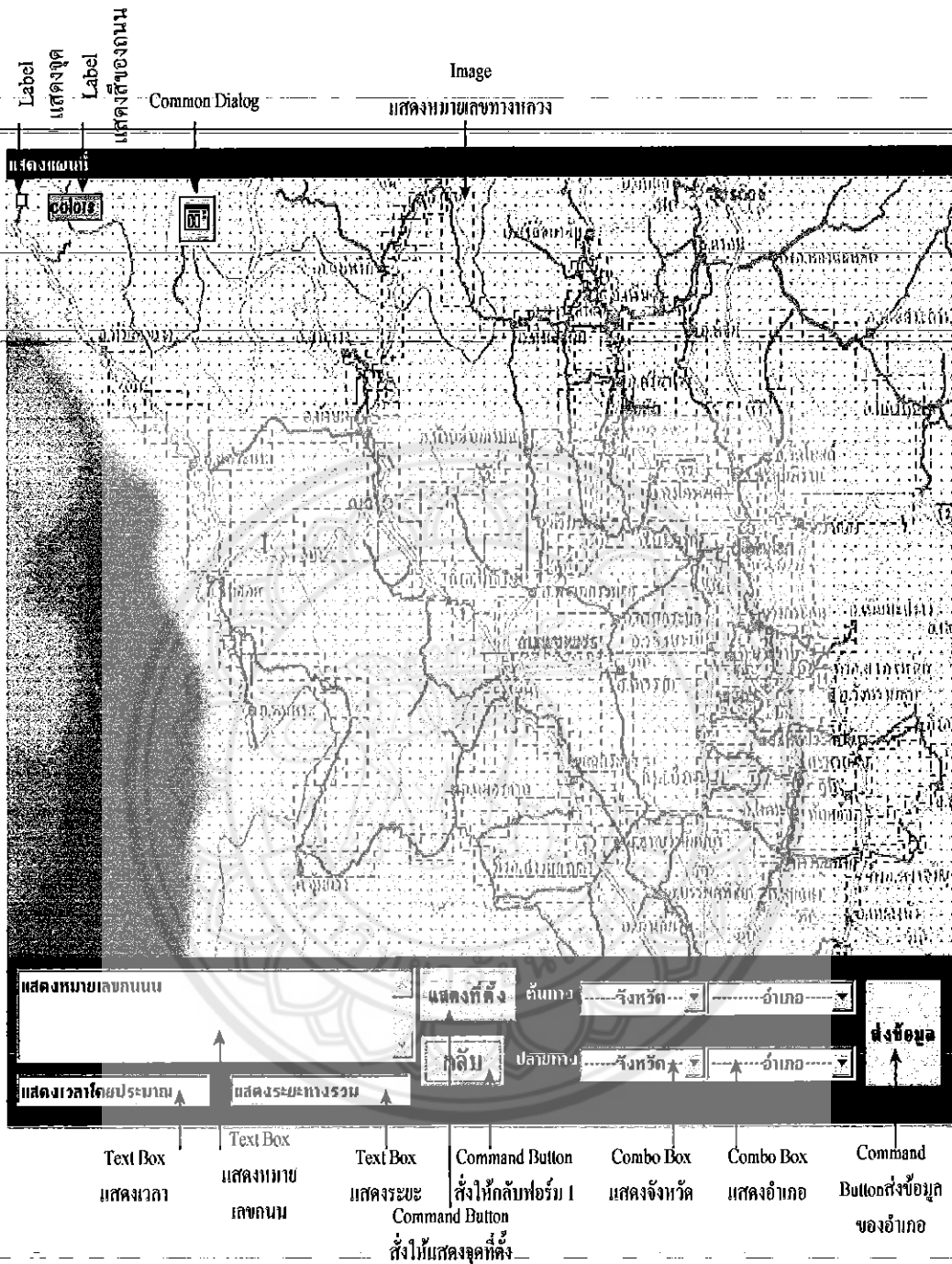
ในฟอรัมที่ 2 จะประกอบไปด้วย 1. Label ของจุด, ถังของถนน, ต้นทาง, ปลายทาง, Text Box แสดงระยะทาง, เวลา, หมายเลขทางหลวง, Image เพื่อแสดงเส้นทางหลวง, Combo Box แสดงอำเภอในจังหวัดนั้นๆ และ Command Button ส่งข้อมูล, แสดงที่ตั้ง, กลับไปฟอรัมที่ 1 ในฟอรัมที่ 2 นี้จะมีการกำหนดค่าพรีอเพอร์ติ์ที่สำคัญ ตาม ตารางที่ 3.2

ตารางที่ 3.2 ค่าพรีอเพอร์ติ์ของฟอรัมที่ 2

Appearance	0-Flat
BorderStyle	1-Fixed single
Picture	Bitmap (กำหนดภาพเป็นพื้นหลัง)
StartPosition	0-Manual



รูปที่ 3.1 รูปแบบของฟอร์มที่ 1



รูปที่ 3.2 รูปแบบของฟอร์มที่ 2

### 3.1.2 ออกแบบและพัฒนาการจัดเก็บฐานข้อมูลเส้นทาง

การจัดเก็บข้อมูลเส้นทาง จากแต่เดิมที่ศึกษานั้นโปรแกรมสามารถทำงานโดยเดินเส้นทางเดียวเท่านั้น แต่ในความเป็นจริงนั้นถนนจะใช้เดินสองเส้นทาง จึงได้มีการเพิ่มข้อมูลเข้าไปเพื่อให้มีการทำงานโดยเดินสองเส้นทาง ในการจัดเก็บข้อมูลเส้นทางนั้นจะเก็บบนแฟ้มที่สามารถแก้ไขได้โดยใช้

โปรแกรม Notepad โดยในแฟ้มจะมีส่วนประกอบที่ทำหน้าที่แตกต่างกันไป โดยแบ่งเป็นส่วนต่างๆดังนี้

1. กำหนดพื้นที่การแสดงผลระหว่างจุดและเส้นทางที่เชื่อมติดต่อกัน ผ่านทางหน้าจอ จากรูปที่ 3.3 ค่าที่กำหนดไว้คือ 7200,6600 หมายความว่า มีค่าความกว้างและความยาวของการแสดงผล เท่ากับ 7200

และ 6600 ตามลำดับ จากรูปที่ 3.4 ประกอบ

2. แสดงจำนวนจุด จากรูปที่ 3.3 ค่าที่กำหนดไว้คือ 7 จะทำให้มีจำนวนจุดเท่ากับ 7 จุด จากรูปที่ 3.4 ประกอบ

3. แสดงจำนวนเส้นทาง จากรูปที่ 3.3 ค่าที่กำหนดไว้คือ 10 จะทำให้มีจำนวนเส้นทางเท่ากับ 10 เส้นทาง จากรูปที่ 3.4 ประกอบ

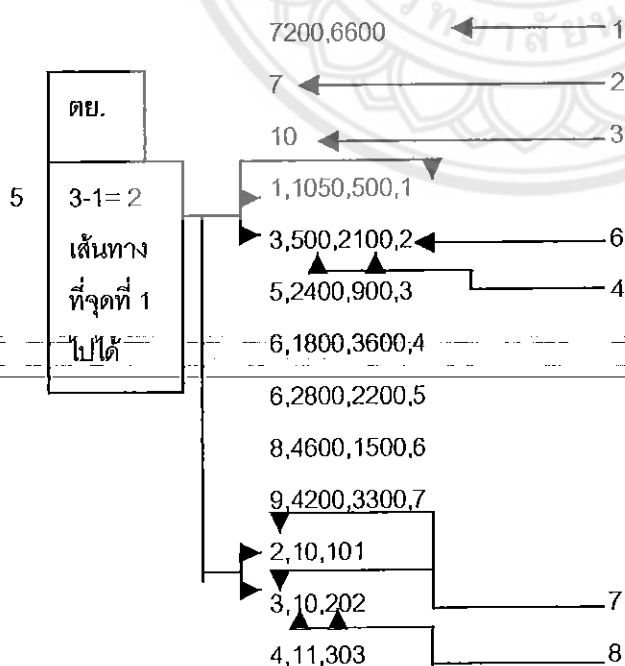
4. แสดงพิกัดของจุด จากรูปที่ 3.3 จะมีพิกัดทั้งหมด 7 พิกัดเท่ากับจำนวนจุดที่กำหนดไว้ จากรูปที่ 3.4 ประกอบ

5. แสดงจำนวนเส้นทางที่จุดนั้นสามารถทำงานได้ จากตัวอย่างในรูปที่ 3.3 จุดที่ 1 จะมีเส้นทาง 2 เส้นทางที่สามารถใช้งานได้ ได้แก่ เส้นทางจากจุดที่ 1 ไปยังจุดที่ 2 และ เส้นทางจากจุดที่ 1 ไปยังจุดที่ 3

6. แสดงหมายเลขของจุด จากรูปที่ 3.3 และรูปที่ 3.4

7. แสดงจุดปลายทางที่จุดนั้นจะไป จากตัวอย่างในรูปที่ 3.3 จุดปลายทางที่จุดที่ 1 จะไปคือ จุดที่ 2 และจุดที่ 3

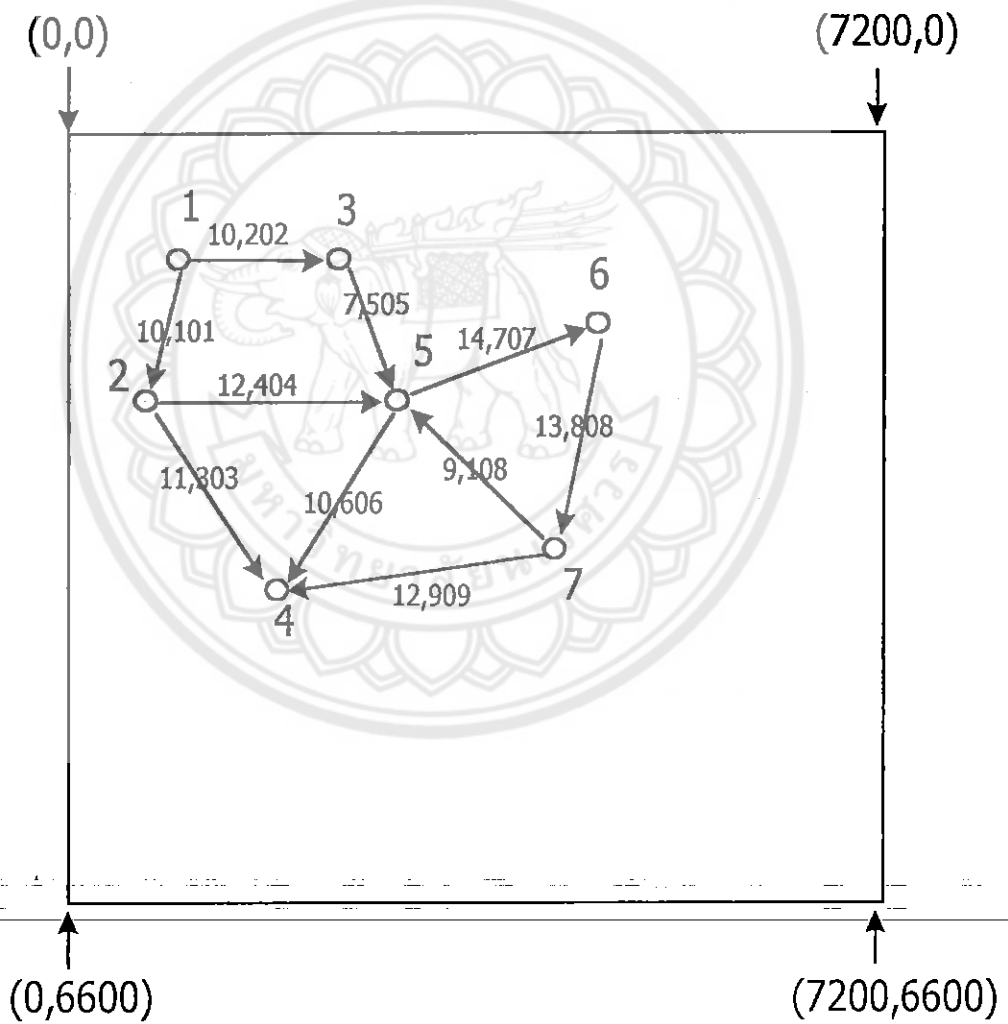
8. แสดงถึงระยะทางและหมายเลขของเส้นทาง จากตัวอย่างในรูปที่ 3.3 เส้นทางจากจุดที่ 1 ไปยังจุดที่ 3 มีระยะทางเท่ากับ 10 และหมายเลขเส้นทางคือ 202 จากรูปที่ 3.4 ประกอบ



รูปที่ 3.3 ส่วนประกอบต่างๆของแฟ้มข้อมูล

5,12,404  
 5,7,505  
 4,10,606  
 6,14,707  
 7,13,808  
 4,12,909  
 5,9,108

รูปที่ 3.3 (ต่อ)

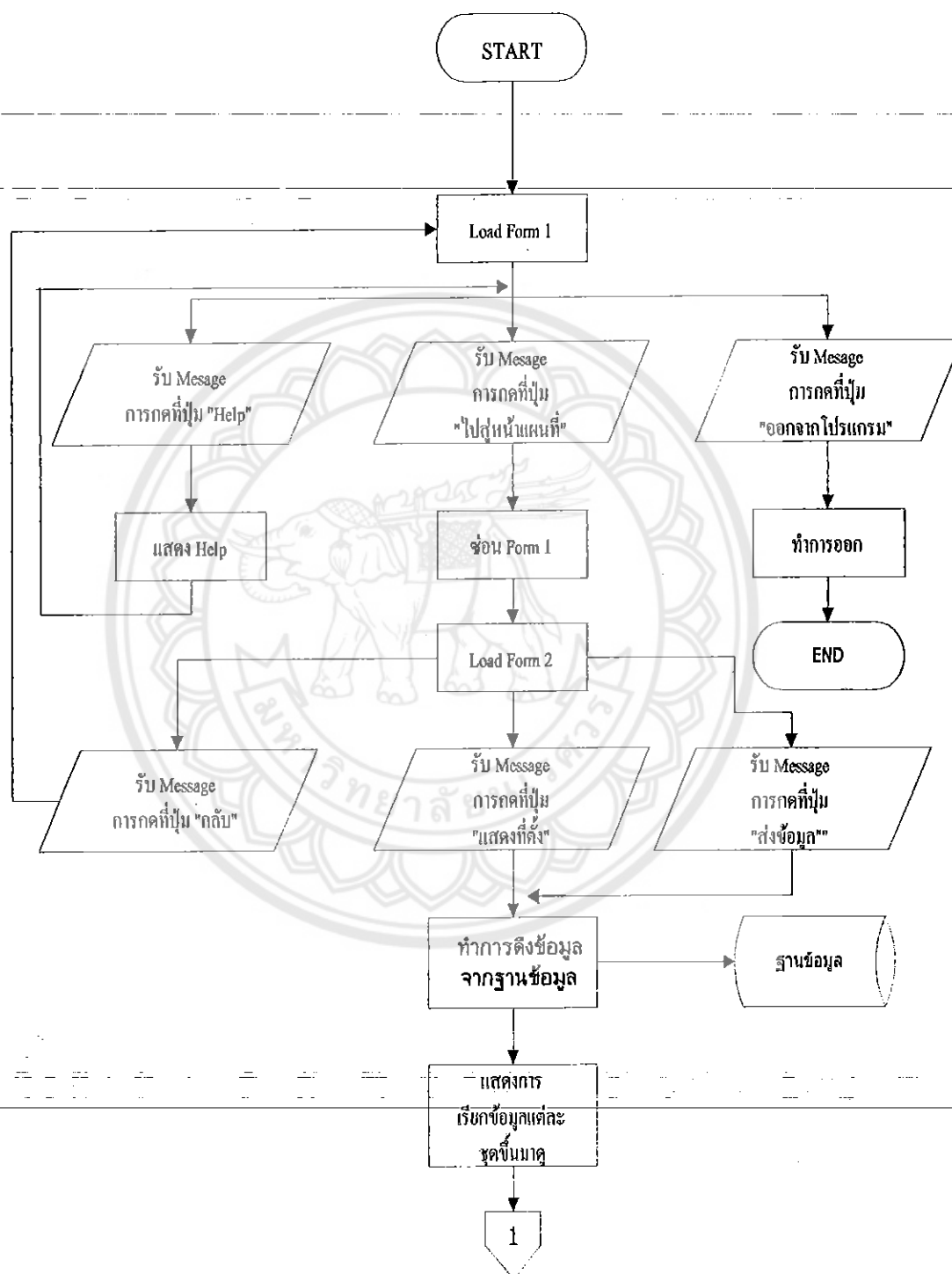


รูปที่ 3.4 ผลลัพธ์เมื่อทำการ Run โปรแกรมจากการจัดเก็บข้อมูล

ข้อสังเกต จากรูปที่ 3.3 ข้อมูลที่เก็บลงไปเป็นการเดินเส้นทางเดียวเท่านั้น

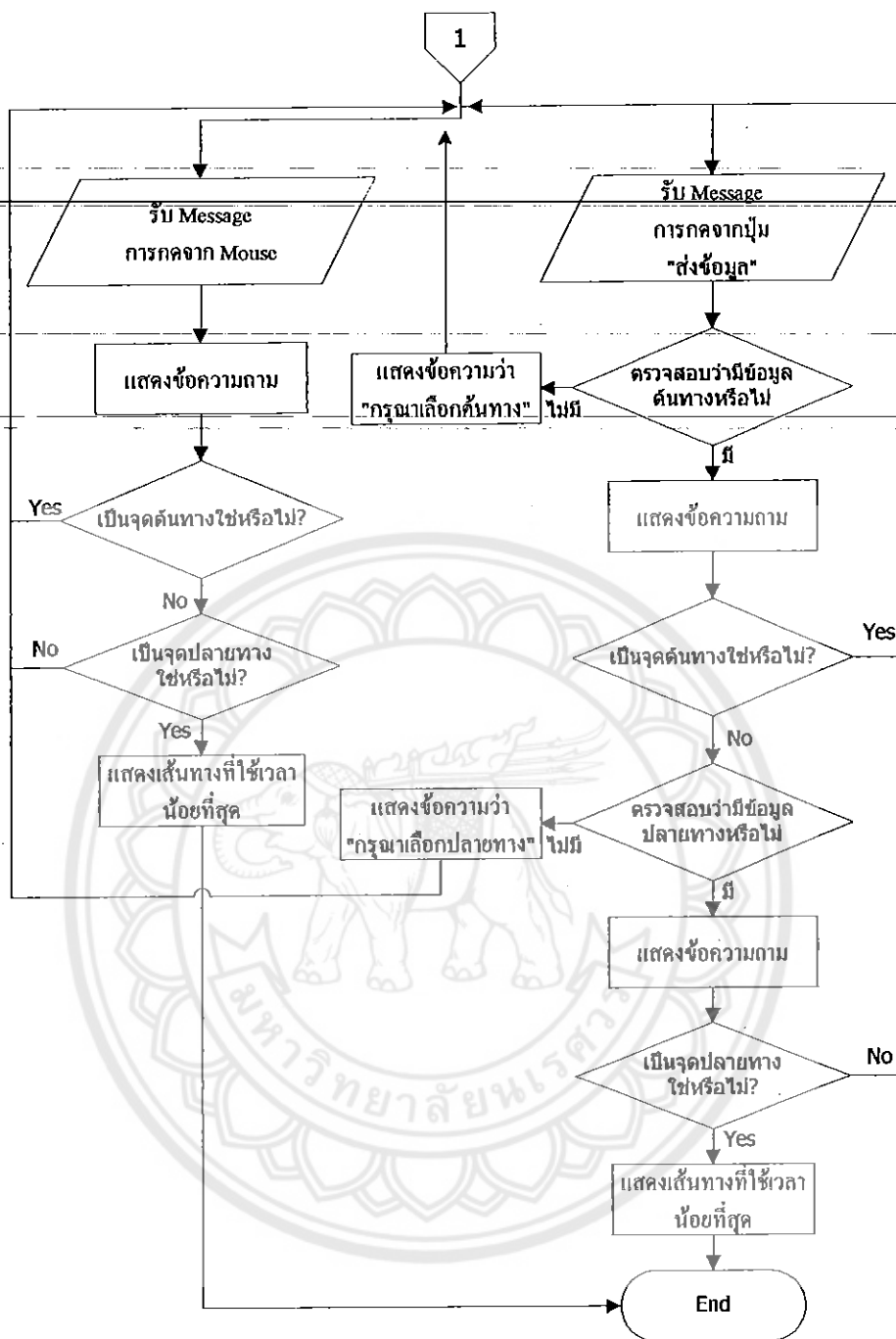
### 3.1.3 ออกแบบและพัฒนาโปรแกรม ในการค้นหาเส้นทางที่ใช้เวลาน้อยที่สุด

ในส่วนของการออกแบบนั้น จะใช้ Flow Chart แสดงหลักการทำงานและเงื่อนไขของโปรแกรมโดยรวม โดยจะมีแผนผังงานดังนี้



รูปที่ 3.5 แผนผังงานโดยรวมของโปรแกรม





รูปที่ 3.5 (ต่อ)

ในส่วนของการพัฒนาโปรแกรมจะมีส่วนต่างๆที่จะพัฒนามีดังนี้

1. การ load ข้อมูลจาก เพิ่มข้อมูลชื่อ NET1.PTH
2. แสดงการ เรียกข้อมูลแต่ละชุดขึ้นมาดู
3. การกำหนดขนาดของ การแสดงผลทางหน้าจอ
4. การรับข้อมูล ที่แสดง จำนวนจุด ทั้งหมด และจำนวน link ทั้งหมด

5. การรับข้อมูลที่แสดง เลขที่จุด (คั่นทาง) พิกัด  $x, y$  และชื่อของจุด ที่ละจุด
6. การรับข้อมูลที่แสดง หมายเลข node (ปลายทาง) ระยะทาง หมายเลขถนน ที่ละจุด
7. การให้ค่าความสำคัญ ของหมายเลขถนน เทียบความเร็ว เพื่อที่จะคำนวณหา เส้นทาง ที่ใช้

ระยะเวลาที่น้อยที่สุด

8. การวาดเส้นทางเชื่อมต่อของข้อมูล โดยมีการวาดหัวลูกศรเพื่อแสดงทิศทางการเดินของจุด
- ซึ่งในการพัฒนานั้นจะใช้คำสั่งที่ได้ศึกษามา เช่น คำสั่งสร้างเส้น, คำสั่งสร้างข้อมูลแบบ

Array, คำสั่งเงื่อนไข เป็นต้น



44 00 594

G

2376.P2

2322/

244C.1

## บทที่ 4

### การทดสอบและวิเคราะห์โปรแกรม

ในบทนี้จะกล่าวถึงลักษณะวิธีการใช้งาน วิธีการทดสอบและวิเคราะห์ผลที่ได้ของโปรแกรม แสดงแผนที่ (Electronic Map)

#### 4.1 ลักษณะทั่วไปของโปรแกรม

โปรแกรมแสดงแผนที่ (Electronic Map) เพื่อที่จะอำนวยความสะดวกแก่ผู้ใช้เส้นทางการจราจรทางบก ได้เลือกให้เส้นทางการจราจรทางบกเพื่อเดินทางไปสู่จุดหมายปลายทางได้ตามความต้องการ เริ่มต้นด้วยการเปิดหน้าจอแรกขึ้นมา โดยหน้าจอแรกจะเป็นส่วนของการเชื่อมโยงไปที่หน้าต่อไป ซึ่งเป็นหน้าจอที่แสดงแผนที่ และหน้าจอแรกนี้มีส่วนของตัวช่วย (Help) ซึ่งจะเป็นส่วนที่แสดงวิธีการทำงานในกรณีที่ผู้ใช้ไม่เข้าใจวิธีการใช้งานโปรแกรม

ลักษณะของโปรแกรมนี้อาจเป็น 2 ส่วน ได้แก่

##### 4.1.1 ส่วนที่รับข้อมูลผ่านการชี้เมาส์ (Mouse) บนแผนที่

ในส่วนนี้ผู้ใช้จะสามารถเลือกสถานที่ รายละเอียดที่แสดงในส่วนนี้ประกอบไปด้วย หมายเลขทางหลวง ชื่อสถานที่ ที่ตั้งของสถานที่ และทางแยก

##### 4.1.2 ส่วนที่รับข้อมูลผ่านทางกรอกข้อมูล

ในส่วนนี้ผู้ใช้จะสามารถเลือกสถานที่ รายละเอียดที่แสดงในส่วนนี้จะมีเพียงชื่อของสถานที่เท่านั้น

#### 4.2 วิธีการใช้งานของโปรแกรม

รูปแบบของโปรแกรมจะมี 2 หน้าจอ

1. ผู้ใช้จะพบหน้าจอแรกโดยหน้าจอแรกจะเป็นส่วนของการเชื่อมโยงไปที่หน้าต่อไป ซึ่งเป็นหน้าจอที่แสดงแผนที่ และหน้าจอแรกนี้มีส่วนของตัวช่วย (Help) ซึ่งจะเป็นส่วนที่แสดงวิธีการทำงานในกรณีที่ผู้ใช้ไม่เข้าใจวิธีการใช้งานโปรแกรม โดยผู้ใช้จะใช้งานโดยการเลือกจากปุ่ม (Command

button) หน้าจอแสดงแผนที่ เมื่อผู้ใช้ได้กดปุ่มนี้แล้ว จะทำการเชื่อมโยงไปยังหน้าต่อไป

2. หน้าจอที่สอง จัดเป็นหน้าจอที่รับการติดต่อกับผู้ใช้ โดยมีการรูปแบบการติดต่อกับผู้ใช้แบ่งออกเป็น 2 ส่วน ได้แก่

##### 2.1 การรับข้อมูลผ่านการชี้เมาส์ (Mouse) บนแผนที่ โดยมีขั้นตอนการทำงานดังนี้

ผู้ใช้กดบนแผนที่บริเวณกรอบสี่เหลี่ยมเล็กๆที่อยู่บริเวณทางแยกหรือตัวอำเภอ โดยผู้ใช้จะต้องทำการกดครั้งแรกเพื่อเลือกเส้นทางว่าผู้ใช้ต้องการเดินทางจากเส้นทางใด ในส่วนของโปรแกรมจะถามอีก

ครั้งเพื่อการยืนยันตำแหน่งต้นทาง ต่อจากนั้นผู้ใช้อาจกำหนดจุดปลายทางจากการกดอีกครั้ง ส่วนของโปรแกรมจะถามอีกครั้งว่าตำแหน่งที่กดครั้งนี้เป็นตำแหน่งต้นทางหรือไม่ ผู้ใช้จะต้องตอบว่า “ไม่ใช่” โปรแกรมจะถามอีกว่าการกดครั้งที่สองนี้เป็นตำแหน่งปลายทางหรือไม่ ถ้าผู้ใช้ตอบว่าใช่ส่วนของโปรแกรมจะแสดงแผนที่ที่เป็นเส้นทางการเดินทางที่ดีที่สุดและเร็วที่สุดกลับมา รวมถึงระยะเวลาโดยประมาณ

### 2.2 การรับข้อมูลผ่านทางกรอกข้อมูล โดยมีขั้นตอนการทำงานดังนี้

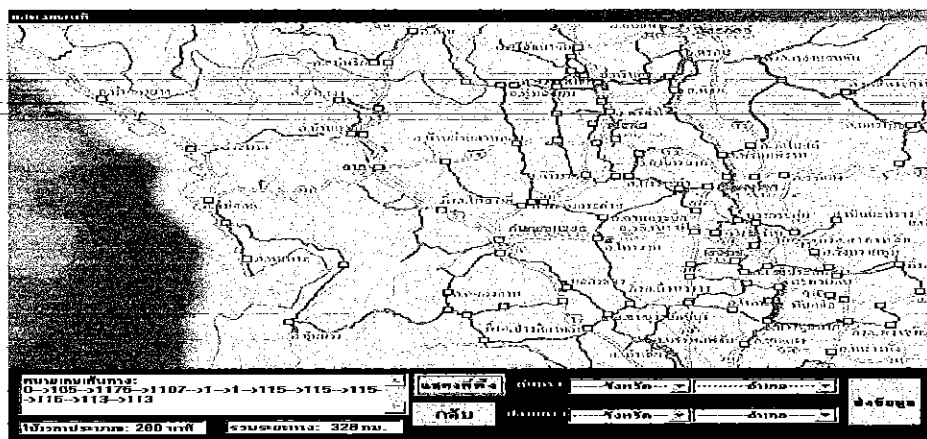
ผู้ใช้เลือกการใช้งานจากการกรอกข้อมูลจังหวัดต้นทาง อำเภอต้นทาง จังหวัดปลายทาง อำเภอปลายทาง โดยการเลือกจากจังหวัดต้นทางอำเภอต้นทาง จังหวัดปลายทางอำเภอปลายทางที่มีให้ หรือผู้ใช้กรอกข้อมูลต้นทางปลายทางเองเมื่อผู้ใช้เลือกข้อมูล จังหวัดต้นทางอำเภอต้นทาง จังหวัดปลายทาง อำเภอปลายทาง แล้วผู้ใช้กดปุ่มส่งข้อมูล เพื่อให้โปรแกรมประมวลผล จากนั้นโปรแกรมจะแสดงผลลัพธ์ออกมาเป็นเส้นทางในการเดินทาง ผลรวมของระยะทางในการเดินทางครั้งนี้ และระยะเวลาในการเดินทางโดยประมาณ

### 4.3 วิธีการทดสอบ

การทดสอบโปรแกรมนี้จะแบ่งการทดสอบ โปรแกรมออกเป็นสองส่วนตามที่ได้กล่าวไว้ข้างต้น

#### 4.3.1 ส่วนของการกดเมาส์

- วิธีการทดสอบ โปรแกรมส่วนของการกดเมาส์ จะเริ่มโดยการกดที่จุดสี่เหลี่ยมเล็กๆ ที่เป็นที่ตั้งของอำเภอ โดยการทดสอบนี้จะเลือกอำเภอทำสองยาง จังหวัดตากเป็นจุดต้นทาง และเลือกอำเภอตะพานหิน จังหวัดพิจิตร เป็นจุดปลายทาง
- ผลการทดสอบ โปรแกรมส่วนของการกดเมาส์ จากการทดสอบผู้ใช้เลือกอำเภอทำสองยางเป็นจุดต้นทาง และอำเภอตะพานหินเป็นจุดปลายทาง จะแสดงผลดังนี้



รูปที่ 4.1 ผลจากการทดสอบในส่วนของการกดเมาส์

- วิเคราะห์ผลการทดสอบ โปรแกรมในส่วนของการกวดมาส์

จากรูปที่ 4.1 จะเป็นผลของการทำงานในส่วนของการรับข้อมูลจากการกวดมาส์ โดยส่วนนี้จะแสดงเส้นทาง ระยะทาง เวลาที่ใช้ในการเดินทาง และหมายเลขทางหลวงที่ผ่าน

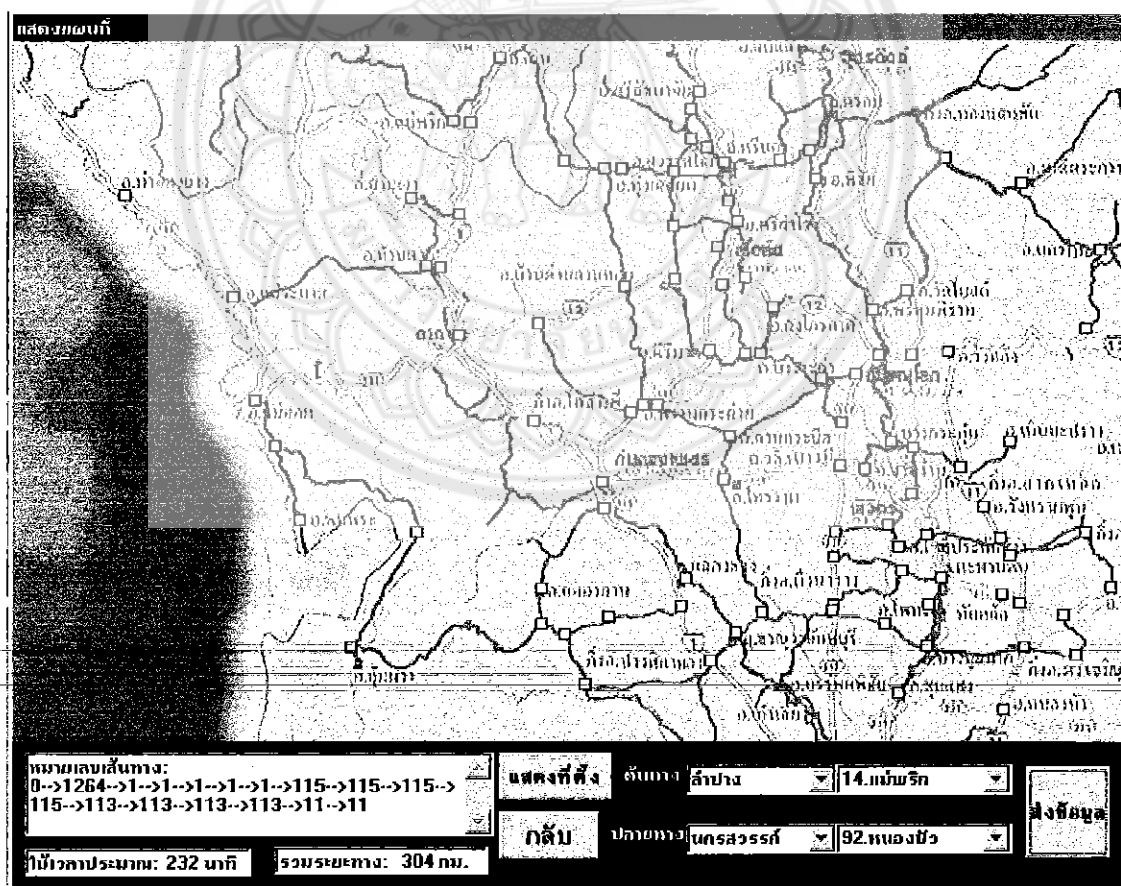
4.3.2 ส่วนของการการรับข้อมูลผ่านทางกรอกข้อมูล

- วิธีการทดสอบโปรแกรมส่วนของการรับข้อมูลผ่านทางกรอกข้อมูล

ผู้ใช้เลือกการใช้งานจากการกรอกข้อมูลจังหวัดต้นทาง อำเภอต้นทาง จังหวัดปลายทาง อำเภอปลายทาง โดยการเลือกจากจังหวัดต้นทางอำเภอต้นทาง จังหวัดปลายทางอำเภอปลายทางที่มีให้ หรือผู้ใช้กรอกข้อมูลต้นทางปลายทางเองเมื่อผู้ใช้เลือกข้อมูล จังหวัดต้นทางอำเภอต้นทาง จังหวัดปลายทาง อำเภอปลายทาง แล้วผู้ใช้กดปุ่มส่งข้อมูล เพื่อให้โปรแกรมประมวลผล โดยในที่นี้จะเลือกอำเภอแม่พริก จังหวัดลำปาง เป็นจุดต้นทาง และเลือกอำเภอหนองบัว จังหวัดนครสวรรค์ เป็นจุดปลายทาง

- ผลการทดสอบโปรแกรมส่วนของการรับข้อมูลผ่านทางกรอกข้อมูล

จากการทดสอบผู้ใช้เลือกอำเภอแม่พริกเป็นจุดต้นทาง และอำเภอหนองบัวเป็นจุดปลายทาง จะแสดงผลดังนี้



รูปที่ 4.2 ผลจากการทดสอบในส่วนของการรับข้อมูลผ่านทางกรอกข้อมูล

- วิเคราะห์ผลการทดสอบโปรแกรมในส่วนของการกวดเมาส์
- จากรูปที่ 4.2 จะเป็นผลของการทำงานในส่วนของการรับข้อมูลจากการกวดเมาส์ โดยส่วนนี้จะแสดงเส้นทาง ระยะทาง เวลาที่ใช้ในการเดินทาง และหมายเลขทางหลวงที่ผ่าน



## บทที่ 5

### บทสรุป

ในบทนี้จะกล่าวถึงการสรุปผลที่ได้จากการทำโครงการ โปรแกรมแสดงผลที่รวมทั้งปัญหาที่เกิดขึ้นพร้อมทั้งวิธีการในการแก้ปัญหาตลอดจนข้อดีและข้อเสียและแนวทางในการพัฒนาในโอกาสต่อไป

#### 5.1 สรุปผล

จากผลการทดสอบ โปรแกรมแสดงผลที่ที่ได้แสดงในบทที่ผ่านมาพบว่ามีโปรแกรมสามารถแสดงผลออกมาได้ตรงตามวัตถุประสงค์ที่ตั้งไว้

#### 5.2 ปัญหาและวิธีการในการแก้ปัญหา

1. ปัญหาด้านการเขียน โปรแกรม ปัญหาในด้านนี้เกิดขึ้นได้ตลอดเวลา วิธีแก้คือ โดยการปรึกษาอาจารย์ที่ปรึกษา หรือค้นคว้าจากตำราต่างๆ
2. ปัญหาด้านการภาพ ภาพไม่คมชัด วิธีแก้คือ ใช้โปรแกรม Adobe Photoshop เข้าช่วยให้ภาพคมชัดขึ้น

#### 5.2 แนวทางในการพัฒนาโปรแกรมต่อไป

การพัฒนาในโอกาสต่อไปมีดังนี้

1. ควรเพิ่มจำนวนสถานที่ อำเภอ และจังหวัด ให้ครอบคลุมทั่วประเทศ
2. ความสามารถของ โปรแกรมในการขยายขนาดของแผนที่ได้

ภาคผนวก ก

คู่มือการติดตั้งโปรแกรมแสดงแผนที่





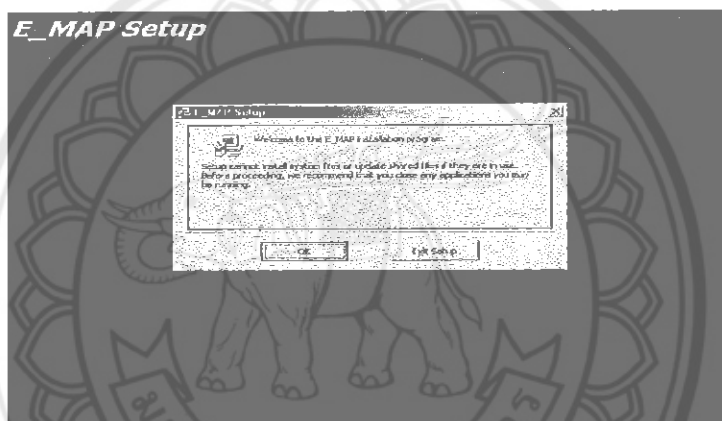
## คู่มือการติดตั้งโปรแกรมแสดงแผนที่

### 1. ความต้องการ

1. มีพื้นที่ในฮาร์ดดิสก์ไม่น้อยกว่า 8-เมกกะไบต์
2. มีระบบปฏิบัติการ Windows 95 หรือสูงกว่า
3. มี Ram ไม่น้อยกว่า 32 เมกกะไบต์
4. ความเร็ว CPU 166 เมกกะเฮิร์ต หรือสูงกว่า

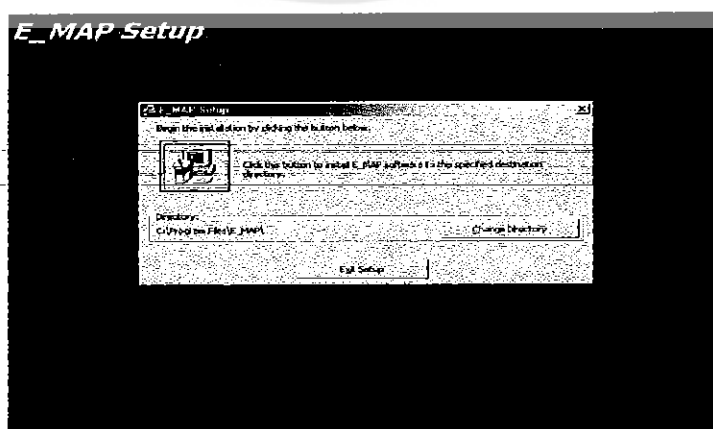
### 2. การติดตั้งโปรแกรมแสดงแผนที่ (Electronic Map software application installation)

1. เลือกเปิดไฟล์จาก CD ROM เลือก Folder ที่ชื่อ E\_MAP
2. ดับเบิล-คลิกที่ไฟล์ setup
3. ทำตามขั้นตอนโดยคลิก ok



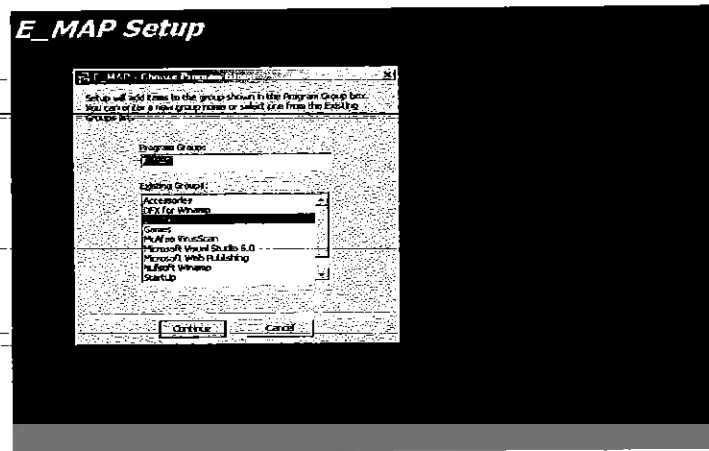
รูปที่ ก.1 การติดตั้งโปรแกรมแสดงแผนที่

### 4. เลือก Directory แล้วคลิก button



รูปที่ ก.2 การเลือก Directory

4. เลือก Program Group แล้วคลิก Continue เป็นอันว่าติดตั้งเสร็จเรียบร้อย



รูปที่ ก.3 การเลือก Program Group



ภาคผนวก ข

รูปแบบโปรแกรมแสดงแผนที่



```
'*****'
```

```
'รูปแบบโปรแกรมการแสดงผลที่
```

```
'*****'
```

```
'PATH_C.FRM
```

```
'*****'
```

```
Option Explicit
```

```
Dim BackX As Single 'ใช้ในการเขียนหัวลูกศร
```

```
Dim BackY As Single
```

```
Dim root As Integer 'การกำหนดจุดเริ่มต้นของการจัดเก็บข้อมูล
```

```
Dim PathToNode As Integer 'การกำหนดการเดินของ node
```

```
'*****'
```

```
'การวาดเส้นทางเชื่อมต่อของข้อมูล โดยมีการวาดหัวลูกศร
```

```
'เพื่อแสดงทิศทางการเดินของจุด
```

```
'*****'
```

```
Private Sub DisplayNetwork()
```

```
Dim node As Integer
```

```
Dim link As Integer
```

```
Dim x1 As Single
```

```
Dim y1 As Single
```

```
Dim x2 As Single
```

```
Dim y2 As Single
```

```
Dim Dx As Single
```

```
Dim Dy As Single
```

```
Dim length As Single
```

```
'แสดงการเชื่อมต่อของเส้นทางต่างๆ.
```

```
    If PathToNode > 0 Then
```

```
        ForeColor = ColorLabel.BackColor
```

```
        DrawWidth = 3 ' วาดเส้นทาง
```

```
        node = PathToNode
```

```
        Do While Parent(node) > 0
```

```

    DrawLink Parent(node), node
    node = Parent(node)
Loop
ForeColor = ColorLabel.Visible
DrawWidth = 1
timerLabel.text = "ใช้เวลาประมาณ:" & Str$(Timer(PathToNode)) & " นาที"
Roadlabel.text = "หมายเลขเส้นทาง: " & Format$(Showroad(PathToNode))
Distancelabel.text = "รวมระยะทาง: " & Str$(Distance(PathToNode)) & " กม."
Else
timerLabel.text = "แสดงเวลาโดยประมาณ"
Roadlabel.text = "แสดงหมายเลขถนน"
Distancelabel.text = "แสดงระยะทางรวม"
End If
End Sub

' *****
' การวาดหัวลูกศร
' *****

Private Sub DrawLink(from_node As Integer, to_node As Integer)
Dim x1 As Single
Dim y1 As Single
Dim x2 As Single
Dim y2 As Single
Dim Dx As Single
Dim Dy As Single
Dim length As Single
x1 = NodeX(from_node)
y1 = NodeY(from_node)
x2 = NodeX(to_node)
y2 = NodeY(to_node)
' วาดหัวลูกศร
Line (x1, y1)-(x2, y2)

```

```
Dx = x2 - x1
```

```
Dy = y2 - y1
```

```
length = Sqr(Dx * Dx + Dy * Dy)
```

```
Dx = Dx / length * BackX
```

```
Dy = Dy / length * BackY
```

```
x2 = x2 - Dx
```

```
y2 = y2 - Dy
```

```
Line (x2, y2)-Step(Dy - Dx, -Dx - Dy)
```

```
Line (x2, y2)-Step(-Dy - Dx, Dx - Dy)
```

```
End Sub
```

```
Private Sub cmbfrom1_Click()
```

```
*****
```

```
' การกำหนดการเลือกจังหวัดค้นหา จากการกรอกข้อมูล จังหวัด
```

```
' ซึ่งจะแสดง อำเภอ ของ จังหวัด นั้นในเวลาต่อมา
```

```
*****
```

```
If cmbfrom1.text = "ตาก" Then
```

```
  cmbfrom2.Clear
```

```
  cmbfrom2.text = "9 .ตาก"
```

```
  With cmbfrom2
```

```
    .AddItem "2 .แม่ระมาด"
```

```
    .AddItem "3 .แม่สอด"
```

```
    .AddItem "5 .พบพระ"
```

```
    .AddItem "7 .อุ้มผาง"
```

```
    .AddItem "1 .ท่าสองยาง"
```

```
    .AddItem "10.สามเงา"
```

```
    .AddItem "13.บ้านตาก"
```

```
  End With
```

```
End If
```

```
If cmbfrom1.text = "ลำปาง" Then
```

```
  cmbfrom2.Clear
```

cmbfrom2.text = "14.แม่พริก"

With cmbfrom2

.AddItem "15.เถิน"

End With

End If

If cmbfrom1.text = "กำแพงเพชร" Then

cmbfrom2.Clear

cmbfrom2.text = "23.กำแพงเพชร"

With cmbfrom2

.AddItem "17.โกสัมพีน"

.AddItem "24.คลองลาน"

.AddItem "30.คลองขลุง"

.AddItem "41.ชาณุวรลักษบุรี"

.AddItem "22.พรานกระต่าย"

.AddItem "36.ลานกระบือ"

.AddItem "29.ไพศาลี"

.AddItem "37.ปางศิลาทอง"

End With

End If

If cmbfrom1.text = "สุโขทัย" Then

cmbfrom2.Clear

cmbfrom2.text = "46.สุโขทัย"

With cmbfrom2

.AddItem "21.บ้านด่านลานหอย"

.AddItem "26.ทุ่งเสลี่ยม"

.AddItem "34.คีรีมาศ"

.AddItem "58.ศรีสัชนาลัย"

.AddItem "44.สวรรคโลก"

.AddItem "45.ศรีสำโรง"

.AddItem "60.กงไกรลาศ"

.AddItem "59.ศรีนคร"

End With

End If

If cmbfrom1.text = "พินิจ โลก" Then

cmbfrom2.Clear

cmbfrom2.text = "65.พินิจ โลก"

With cmbfrom2

.AddItem "64.พรหมพิราม"

.AddItem "72.วัด โบสถ์"

.AddItem "82.ชาติตระการ"

.AddItem "83.นครไทย"

.AddItem "66.บางกระทุ่ม"

.AddItem "85.วังทอง"

.AddItem "97.เนินมะปราง"

.AddItem "48.บางระกำ"

End With

End If

If cmbfrom1.text = "พิจิตร" Then

cmbfrom2.Clear

cmbfrom2.text = "75.พิจิตร"

With cmbfrom2

.AddItem "77.โพธิ์ประทับช้าง"

.AddItem "86.สามโก้"

.AddItem "98.วังทรายพูน"

.AddItem "78.ตะพานหิน"

.AddItem "89.ทับคล้อ"

.AddItem "71.โพธิ์ทะเล"

.AddItem "79.บางมูลนาก"

.AddItem "67.บึงนาราง"

.AddItem "91.คองเจริญ"



```
.AddItem "49.วชิรบารมี"
```

```
.AddItem "68.สามง่าม"
```

```
End With
```

```
End If
```

```
If cmbfrom1.text = "เพชรบูรณ์" Then
```

```
    cmbfrom2.Clear
```

```
    cmbfrom2.text = "96.วังโป่ง"
```

```
    With cmbfrom2
```

```
        .AddItem "95.ชนแดน"
```

```
    End With
```

```
End If
```

```
If cmbfrom1.text = "อุดรธานี" Then
```

```
    cmbfrom2.Clear
```

```
    cmbfrom2.text = "63.พิชัย"
```

```
End If
```

```
If cmbfrom1.text = "นครสวรรค์" Then
```

```
    cmbfrom2.Clear
```

```
    cmbfrom2.text = "80.ชุมแสง"
```

```
    With cmbfrom2
```

```
        .AddItem "92.หนองบัว"
```

```
    End With
```

```
End If
```

```
End Sub
```

```
Private Sub cmbto1_Click()
```

```
*****
```

```
' การกำหนดการเลือกจังหวัดปลายทาง จากการกรอกข้อมูล จังหวัด
```

```
' ซึ่งจะแสดง อำเภอ ของ จังหวัด นั้นในเวลาต่อมา
```

```
*****
```

If cmbto1.text = "ตากล" Then

cmbto2.Clear

cmbto2.text = "9 .ตากล"

With cmbto2

.AddItem "2 .แม่ระมาด"

.AddItem "3 .แม่สอด"

.AddItem "5 .พื้บพระ"

.AddItem "7 .อุ้มผาง"

.AddItem "1 .ท่าสองยาง"

.AddItem "10.สามเงา"

.AddItem "13.บ้านตากล"

End With

End If

If cmbto1.text = "ลำปาง" Then

cmbto2.Clear

cmbto2.text = "14.แม่พริก"

With cmbto2

.AddItem "15.เถิน"

End With

End If

If cmbto1.text = "กำแพงเพชร" Then

cmbto2.Clear

cmbto2.text = "23.กำแพงเพชร"

With cmbto2

.AddItem "17.โกสัมพี"

.AddItem "24.คลองลาน"

.AddItem "30.คลองขลุง"

.AddItem "41.ขามเฒ่า"

.AddItem "22.พรานกระต่าย"

```
.AddItem "36.ลานกระบือ"  
.AddItem "29.ไทรงาม"  
.AddItem "37.ปางศิลาทอง"
```

```
End With
```

```
End If
```

```
If cmbto1.text = "สุโขทัย" Then
```

```
    cmbto2.Clear  
    cmbto2.text = "46.สุโขทัย"
```

```
    With cmbto2
```

```
        .AddItem "21.บ้านด่านลานหอย"  
        .AddItem "26.ทุ่งเสลี่ยม"  
        .AddItem "34.คีรีมาศ"  
        .AddItem "58.ศรีตึชนาลัย"  
        .AddItem "44.สวรรคโลก"  
        .AddItem "45.ศรีสำโรง"  
        .AddItem "60.กงไกรลาศ"  
        .AddItem "59.ศรีนคร"
```

```
    End With
```

```
End If
```

```
If cmbto1.text = "พิษณุโลก" Then
```

```
    cmbto2.Clear  
    cmbto2.text = "65.พิษณุโลก"
```

```
    With cmbto2
```

```
        .AddItem "64.พรหมพิราม"  
        .AddItem "72.วัดโบสถ์"  
        .AddItem "82.ชาติตระการ"  
        .AddItem "83.นครไทย"  
        .AddItem "66.บางกระทุ่ม"  
        .AddItem "85.วังทอง"  
        .AddItem "97.เนินมะปราง"
```

```
.AddItem "48.บางระกำ"
```

```
End With
```

```
End If
```

```
If cmbto1.text = "พิจิตร" Then
```

```
    cmbto2.Clear
```

```
    cmbto2.text = "75.พิจิตร"
```

```
With cmbto2
```

```
    .AddItem "77.โพธิ์ประทับช้าง"
```

```
    .AddItem "86.สากเหล็ก"
```

```
    .AddItem "98.วังทรายพูน"
```

```
    .AddItem "78.ตะพานหิน"
```

```
    .AddItem "89.ทับคล้อ"
```

```
    .AddItem "71.โพธิ์ทะเล"
```

```
    .AddItem "79.บางมูลนาก"
```

```
    .AddItem "67.บึงนาราง"
```

```
    .AddItem "91.คงเจริญ"
```

```
    .AddItem "49.วชิรบำรุง"
```

```
    .AddItem "68.สามง่าม"
```

```
End With
```

```
End If
```

```
If cmbto1.text = "เพชรบูรณ์" Then
```

```
    cmbto2.Clear
```

```
    cmbto2.text = "96.วังโป่ง"
```

```
With cmbto2
```

```
    .AddItem "95.ชนแดน"
```

```
End With
```

```
End If
```

```
If cmbto1.text = "อุตรดิตถ์" Then
```

```
    cmbto2.Clear
```

```
cmbto2.text = "63.พีชัย"
```

```
End If
```

```
If cmbto1.text = "นครสวรรค์" Then
```

```
    cmbto2.Clear
```

```
    cmbto2.text = "80.ชุมแสง"
```

```
    With cmbto2
```

```
        .AddItem "92.หนองบัว"
```

```
    End With
```

```
End If
```

```
End Sub
```

```
Private Sub cmdfrom1_Click()
```

```
*****
```

```
' การโหลดข้อมูลจาก .NET1.PTH
```

```
*****
```

```
Dim errno As Integer
```

```
LoadDialog.Filter = ".NET1.PTH"
```

```
LoadDialog.FileName = ".NET1.PTH"
```

```
LoadDialog.InitDir = Right$(LoadDialog.FileName, Len(LoadDialog.FileName) - Len
```

```
(LoadDialog.FileTitle))
```

```
LoadNetwork (LoadDialog.FileName)
```

```
*****
```

```
' การรับข้อมูลจากการกรอกข้อมูลจังหวัดต้นทาง
```

```
' และจังหวัดปลายทาง ซึ่งจะแสดงอำเภอต้นทาง ปลายทาง
```

```
*****
```

```
Dim tontang As Integer
```

```
Dim got As Integer
```

```
Dim plytang As Integer
```

```
Dim One As Integer
```

```
Dim Two As Integer
```

```

If cmbfrom2.text <> "-----อำเภอ-----" Then
    One = changedata(cmbfrom2.text)
    tontang = One
    got = 1
    GetData tontang, got
Else
    MsgBox "กรุณาเลือกต้นทาง", vbOKOnly + vbExclamation, "ไม่เลือกต้นทาง"
Exit Sub
End If

If cmbto2.text <> "-----อำเภอ-----" Then
    Two = changedata(cmbto2.text)
    plytang = Two
    got = 2
    GetData2 plytang, got
Else
    MsgBox "กรุณาเลือกปลายทาง", vbOKOnly + vbExclamation, "ไม่เลือกปลายทาง"
End If
End Sub

Private Sub Command1_Click() 'ปุ่มกลับไปฟอร์มแรก
    PathForm.Hide
    Form1.Show
End Sub

Private Sub Command2_Click()
    *****
    ' การโหลดข้อมูลจาก .\NET1.PTH
    *****
    Dim errno As Integer
    LoadDialog.Filter = ".\NET1.PTH"

```

```

LoadDialog.FileName = ".\NET1.PTH"
LoadDialog.InitDir = Right$(LoadDialog.FileName, Len(LoadDialog.FileName) - Len
(LoadDialog.FileTitle))

```

```
LoadNetwork (LoadDialog.FileName)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
*****
```

```
' แสดง จังหวัดต้นทาง และจังหวัดปลายทาง
```

```
*****
```

```
With cmbfrom1
```

```
.AddItem "ตาก"
```

```
.AddItem "กำแพงเพชร"
```

```
.AddItem "สุโขทัย"
```

```
.AddItem "พิษณุโลก"
```

```
.AddItem "พิจิตร"
```

```
.AddItem "ลำปาง"
```

```
.AddItem "เพชรบูรณ์"
```

```
.AddItem "อุตรดิตถ์"
```

```
.AddItem "นครสวรรค์"
```

```
End With
```

```
With cmbto1
```

```
.AddItem "ตาก"
```

```
.AddItem "กำแพงเพชร"
```

```
.AddItem "สุโขทัย"
```

```
.AddItem "พิษณุโลก"
```

```
.AddItem "พิจิตร"
```

```
.AddItem "เพชรบูรณ์"
```

```
.AddItem "ลำปาง"
```

```
.AddItem "อุตรดิตถ์"
```

```
.AddItem "นครสวรรค์"
```

End With

End Sub

---

Private Sub LoadNetwork(Filename As String)

\*\*\*\*\*

'แสดงการ เรียกข้อมูลแต่ละ ชุคขึ้นมาดู

\*\*\*\*\*

Dim wid As Single

Dim hgt As Single

Dim filenum As Integer

Dim i As Integer

' เปิด file.

On Error GoTo OpenError

filenum = FreeFile

Open Filename For Input As filenum

For i = 1 To NumNodes

    Unload NodeLabel(i)

Next i

NumNodes = 0

NumLinks = 0

root = 0

PathToNode = 0

Cls

---

\*\*\*\*\*

' การกำหนดขนาดของ from

\*\*\*\*\*

Input #filenum, wid, hgt

Width = wid

Height = hgt

timerLabel.Top = ScaleHeight - 1.5 \* timerLabel.Height



\*\*\*\*\*

' การรับข้อมูล ที่แสดง จำนวน node ทั้งหมด และจำนวน link ทั้งหมด

\*\*\*\*\*

On Error GoTo ReadError

Input #filenum, NumNodes, NumLinks

wid = NodeLabel(0).Width

hgt = NodeLabel(0).Height

BackX = wid / 2

BackY = hgt / 2

ReDim FirstLink(1 To NumNodes + 1) ' Extra for sentinel.

ReDim NodeX(1 To NumNodes)

ReDim NodeY(1 To NumNodes)

ReDim Parent(1 To NumNodes)

ReDim Timer(1 To NumNodes)

ReDim name(1 To NumNodes)

ReDim Showroad(1 To NumNodes)

ReDim Distance(1 To NumNodes)

For i = 1 To NumNodes

\*\*\*\*\*

' การรับข้อมูล ที่แสดง เลขที่ node(ต้นทาง) พิกัด x,y และชื่อของ node ที่ละ node

\*\*\*\*\*

Input #filenum, FirstLink(i), NodeX(i), NodeY(i), name(i)

Load NodeLabel(i)

NodeLabel(i).Move NodeX(i) - wid / 2, NodeY(i) - hgt / 2

NodeLabel(i).Visible = True

Next i

FirstLink(NumNodes + 1) = NumLinks + 1

ReDim Cost(1 To NumLinks)

ReDim road(1 To NumLinks)

ReDim Typeroad(1 To NumLinks)

ReDim time(1 To NumLinks)

ReDim Tonode(1 To NumLinks)

```
For i = 1 To NumLinks
```

```
*****
```

```
' การรับข้อมูลที่แสดง หมายเลข_node(ปลายทาง) ระยะทาง หมายเลขถนน ที่ละ node
```

```
*****
```

```
Input #filenum, Tonode(i), Cost(i), road(i)
```

```
*****
```

```
' การให้ค่าความสำคัญ ของหมายเลขถนน เทียบความเร็ว
```

```
' เพื่อที่จะคำนวณหาเส้นทาง ที่ใช้เวลาน้อยที่สุด
```

```
*****
```

```
Typeroad(i) = road(i)
```

```
Typeroad(i) = Len(Typeroad(i))
```

```
Select Case Typeroad(i)
```

```
Case 1
```

```
Typeroad(i) = Cost(i) * 0.67 'ความเร็ว 90 km/hr
```

```
time(i) = Typeroad(i)
```

```
Case 2
```

```
Typeroad(i) = Cost(i) * 0.75 'ความเร็ว 80 km/hr
```

```
time(i) = Typeroad(i)
```

```
Case 3
```

```
Typeroad(i) = Cost(i) * 0.86 'ความเร็ว 70 km/hr
```

```
time(i) = Typeroad(i)
```

```
Case 4
```

```
Typeroad(i) = Cost(i) * 1 'ความเร็ว 60 km/hr
```

```
time(i) = Typeroad(i)
```

```
End Select
```

```
Next i
```

```
' ปิด file.
```

```
On Error GoTo CloseError
```

```
Close #filenum
```

```
Refresh
```

Exit Sub

-----  
OpenError:

Beep

MsgBox "Error opening file "" & Filename & ""." & Chr\$(13) & Chr\$(10) & Error\$

Exit Sub

-----  
ReadError:

Beep

MsgBox "Error reading from file "" & Filename & ""." & Chr\$(13) & Chr\$(10) & Error\$

NumNodes = 0

NumLinks = 0

Exit Sub

CloseError:

Beep

MsgBox "Error closing file "" & Filename & ""." & Chr\$(13) & Chr\$(10) & Error\$

Exit Sub

End Sub

Private Sub mnuFileExit\_Click()

End

End Sub

-----  
Private Sub mnuFileLoad\_Click()

\*\*\*\*\*

' การ load ข้อมูลจาก ANET1.PTH

\*\*\*\*\*

Dim errno As Integer

LoadDialog.Filter = ".NET1.PTH"

LoadDialog.FileName = ".NET1.PTH"

```

LoadDialog.InitDir = Right$(LoadDialog.FileName, Len(LoadDialog.FileName) - Len
(LoadDialog.FileTitle))
LoadNetwork (LoadDialog.FileName)

```

```
End Sub
```

```
Private Sub Form_Paint()
```

```
Call DisplayNetwork
```

```
End Sub
```

```
' *****
```

```
' การกำหนดข้อมูลจากการกรอกข้อมูล ต้นทาง และปลายทาง
```

```
' *****
```

```
Private Sub NodeLabel_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As
Single, Y As Single)
```

```
Dim aa As Integer
```

```
Dim bb As Integer
```

```
Dim i As Integer
```

```
aa = Index
```

```
bb = Index
```

```
If MsgBox("'" & NodeLabel(aa).ToolTipText & " เป็นจุดต้นทางของคุณ ใช่หรือไม่", vbYesNo +
vbQuestion, "กำหนดต้นทาง") = vbYes Then
```

```
Button = 2 Or root < 1
```

```
root = Index ' Save root.
```

```
PathToNode = 0
```

```
On Error GoTo FindPathError
```

```
FindPathTree root ' หา เส้นทาง shortest path tree.
```

```
On Error GoTo 0
```

```
MousePointer = 0
```

```
Else
```

```
If MsgBox("" & "แต่" & NodeLabel(bb).ToolTipText & " เป็นจุดปลายทางของคุณ ใช่หรือไม่",
vbYesNo + vbExclamation, "กำหนดปลายทาง") = vbYes Then
```

```
PathToNode = Index
```

```
Call DisplayNetwork
```

```
End If
```

```
End If
```

```
Refresh ' วาดหน้าจอใหม่
```

```
Exit Sub
```

```
FindPathError:
```

```
MousePointer = 0
```

```
Beep
```

```
MsgBox "Error finding shortest path tree." & Error$
```

```
Exit Sub
```

```
End Sub
```

```
Public Sub GetData(Index As Integer, Button As Integer)
```

```
*****
```

```
' การรับข้อมูล จากการกรอกข้อมูลต้นทาง
```

```
*****
```

```
If MsgBox("ต้นทางคือจังหวัด " & cmbfrom1.text & " " & cmbfrom2.text & " ใช่หรือไม่",
vbYesNo + vbQuestion, "กำหนดต้นทาง") = vbYes Then
```

```
Button = 2 Or root < 1
```

```
root = Index
```

```
PathToNode = 0
```

```
FindPathTree root
```

```
On Error GoTo 0
```

```
End If
```

```
Refresh
```

```
End Sub
```

```
Public Sub GetData2(Index As Integer, Button As Integer)
```

```
*****
```

```
' การรับข้อมูล จากการกรอกข้อมูลปลายทาง
```

```
*****
```

```
If MsgBox("ปลายทางคือ- "&cmbto1.text &" "&cmbto2.text &" ใช่หรือไม่", vbYesNo +  
vbQuestion, "กำหนดปลายทาง") = vbYes Then
```

```
    PathToNode = Index
```

```
    Call DisplayNetwork
```

```
    End If
```

```
End Sub
```

```
Private Sub NodeLabel_Mousemove(Index As Integer, Button As Integer, Shift As Integer, X As  
Single, Y As Single)
```

```
*****
```

```
' การกำหนด ชื่อของ Node ต่างๆ
```

```
*****
```

```
NodeLabel(1).ToolTipText = "ท่าสองยาง"
```

```
NodeLabel(2).ToolTipText = "แม่ระมาด"
```

```
NodeLabel(3).ToolTipText = "แม่สอด"
```

```
NodeLabel(5).ToolTipText = "พบพระ"
```

```
NodeLabel(7).ToolTipText = "อุ้มผาง"
```

```
NodeLabel(9).ToolTipText = "ตาก"
```

```
NodeLabel(10).ToolTipText = "สามเงา"
```

```
NodeLabel(13).ToolTipText = "บ้านตาก"
```

```
NodeLabel(14).ToolTipText = "แม่พริก"
```

```
NodeLabel(15).ToolTipText = "เถิน"
```

```
NodeLabel(17).ToolTipText = "โกสัมพีนี"
```

```
NodeLabel(23).ToolTipText = "กำแพงเพชร"
```

```
NodeLabel(24).ToolTipText = "คลองลาน"
```

```
NodeLabel(37).ToolTipText = "ปางศิลาทอง"
```

```
NodeLabel(30).ToolTipText = "คลองขลุง"
```

NodeLabel(41).ToolTipText = "ขามูวรวลัถษบรู"

NodeLabel(22).ToolTipText = "พรานกระต่าย"

NodeLabel(21).ToolTipText = "บ้านด่านถานหอย"

NodeLabel(26).ToolTipText = "ทุ่งเสถียม"

NodeLabel(34).ToolTipText = "คีรีมาศ"

NodeLabel(58).ToolTipText = "ศรีสัชนาลัย"

NodeLabel(44).ToolTipText = "สวรรคโลก"

NodeLabel(45).ToolTipText = "ศรีสำโรง"

NodeLabel(46).ToolTipText = "สุโขทัย"

NodeLabel(60).ToolTipText = "กงไกรลาศ"

NodeLabel(48).ToolTipText = "บางระกำ"

NodeLabel(36).ToolTipText = "ลานกระบือ"

NodeLabel(29).ToolTipText = "ไทรงาม"

NodeLabel(63).ToolTipText = "พิชัย"

NodeLabel(64).ToolTipText = "พรหมพิราม"

NodeLabel(72).ToolTipText = "วัดโบสถ์"

NodeLabel(82).ToolTipText = "ชาติตระการ"

NodeLabel(83).ToolTipText = "นครไทย"

NodeLabel(85).ToolTipText = "วังทอง"

NodeLabel(65).ToolTipText = "พิษณุโลก"

NodeLabel(49).ToolTipText = "วชิรบารมี"

NodeLabel(68).ToolTipText = "สามง่าม"

NodeLabel(66).ToolTipText = "บางกระทุ่ม"

NodeLabel(75).ToolTipText = "พิจิตร"

NodeLabel(77).ToolTipText = "โพธิ์ประทับช้าง"

NodeLabel(86).ToolTipText = "สากเหล็ก"

NodeLabel(97).ToolTipText = "เนินมะปราง"

NodeLabel(98).ToolTipText = "วังทรายพูน"

NodeLabel(78).ToolTipText = "ตะพานหิน"

NodeLabel(89).ToolTipText = "ทับคล้อ"

NodeLabel(96).ToolTipText = "วังโป่ง"

NodeLabel(95).ToolTipText = "ชนแดน"

```
NodeLabel(91).ToolTipText = "คองเจริญ"
```

```
NodeLabel(92).ToolTipText = "หนองบัว"
```

```
NodeLabel(80).ToolTipText = "ชุมแสง"
```

```
NodeLabel(71).ToolTipText = "โพธิ์ทะเล"
```

```
NodeLabel(79).ToolTipText = "บางมูลนาก"
```

```
NodeLabel(59).ToolTipText = "ศรีนคร"
```

```
NodeLabel(67).ToolTipText = "ปึงนาราง"
```

```
End Sub
```

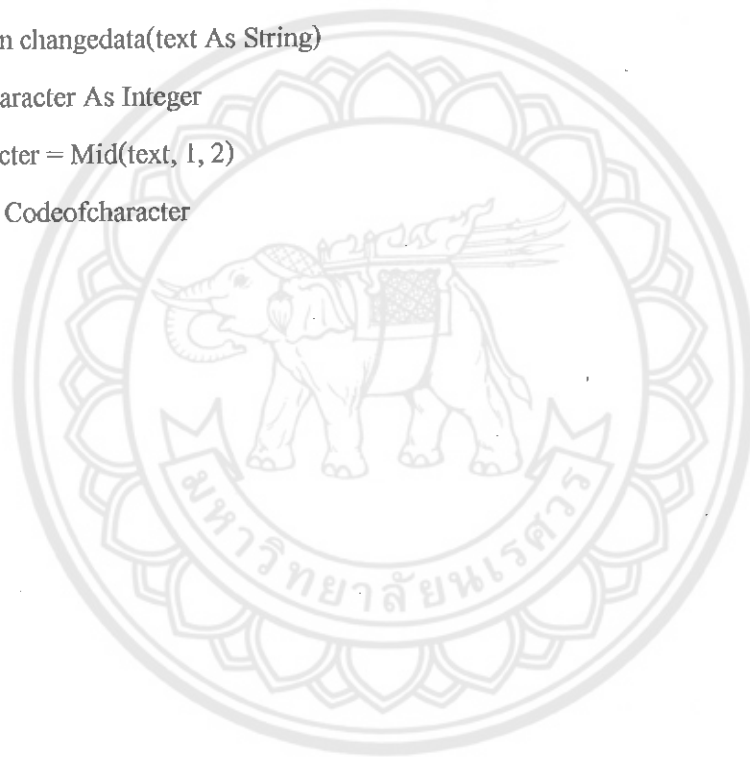
```
Public Function changedata(text As String)
```

```
Dim Codeofcharacter As Integer
```

```
Codeofcharacter = Mid(text, 1, 2)
```

```
changedata = Codeofcharacter
```

```
End Function
```





```
*****  
' form1.frm  
*****
```

```
Private Sub cmdgotomap_Click() 'ปุ่ม ไปฟอร์มแสดงแผนที่
```

```
Form1.Hide
```

```
PathForm.Show
```

```
End Sub
```

```
Private Sub Command1_Click() 'ปุ่มออกจากโปรแกรม
```

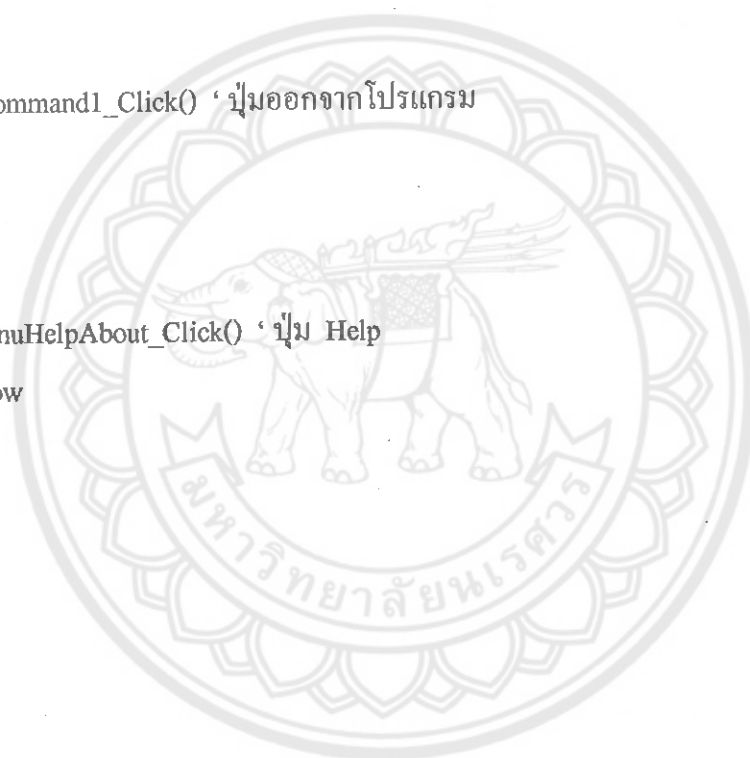
```
End
```

```
End Sub
```

```
Private Sub mnuHelpAbout_Click() 'ปุ่ม Help
```

```
FormHelp.Show
```

```
End Sub
```



```
! *****
```

```
' PATH_C.BAS
```

```
' Source code for label correcting -path program.
```

```
! *****
```

```
Option Explicit
```

```
Global Typeroad() As String
```

```
Global time() As Double
```

```
Global Distance() As Integer
```

```
Global NumNodes As Integer
```

```
Global NumLinks As Integer
```

```
Global FirstLink() As Integer
```

```
Global Cost() As Integer
```

```
Global NodeX() As Single
```

```
Global NodeY() As Single
```

```
Global Tonode() As Integer
```

```
Global Parent() As Integer
```

```
Global Timer() As Integer
```

```
Global Showroad() As String
```

```
Global road() As Integer
```

```
Global name() As String
```

```
! *****
```

```
' Compute the path tree rooted at node root.
```

```
! *****
```

```
Sub FindPathTree(root As Integer)
```

```
Const INFINITY = 32767
```

```
Const END_OF_LIST = 0
```

```
Const NOT_IN_LIST = -1
```

```
Dim CL() As Integer
```

```
Dim last_cl As Integer ' Last item in the list.
```

```
Dim next_cl As Integer ' First item in the list.
```

```
Dim node As Integer
```

```
Dim link As Integer
```

```
Dim to_node As Integer
```

```
' We can have at most NumNodes nodes in the list.
```

```
ReDim CL(1 To NumNodes)
```

```
' Initialize Timer and Parent arrays.
```

```
For node = 1 To NumNodes
```

```
    Parent(node) = 0 ' Not in the tree.
```

```
    Timer(node) = INFINITY ' Best Timer infinite.
```

```
    Showroad(node) = INFINITY
```

```
    Distance(node) = INFINITY
```

```
    CL(node) = NOT_IN_LIST ' Not in candidate list.
```

```
Next node
```

```
Timer(root) = 0 ' The root gets Timerance 0.
```

```
Showroad(root) = 0
```

```
Distance(root) = 0
```

```
' Put the Root on the candidate list.
```

```
next_cl = root ' Oldest item in the list.
```

```
last_cl = root ' Newest item in the list.
```

```
CL(root) = END_OF_LIST
```

' Repeat until the candidate list is empty.

Do While next\_cl  $\diamond$  END\_OF\_LIST

— node = next\_cl —

' Check the links out of this node.

For link = FirstLink(node) To FirstLink(node + 1) - 1

to\_node = Tonode(link)

If Timer(to\_node) > Timer(node) + time(link) Then

Timer(to\_node) = Timer(node) + time(link)

' If this is an improvement for to\_node,

' update its Timer and Parent entries.

Parent(to\_node) = node

Distance(to\_node) = Distance(node)

Distance(to\_node) = Distance(to\_node) + Cost(link)

Showroad(to\_node) = Showroad(node)

Showroad(to\_node) = Showroad(to\_node) & "--->" & road(link)

If CL(to\_node) = NOT\_IN\_LIST Then

CL(last\_cl) = to\_node

last\_cl = to\_node

CL(to\_node) = END\_OF\_LIST

End If

End If

Next link

' Remove the node from the candidate list.

next\_cl = CL(node)

CL(node) = NOT\_IN\_LIST

Loop

End Sub

## เอกสารอ้างอิง

- [1] ตมศักดิ์ ศรีขจรเกียรติ. **Visual-basic 6 ครั้งที่ 4**. กรุงเทพฯ : หจก. บีบีไอไฟล์พับลิชชิง.
- [2] Chanakhan Loetchanwanit. **Highway map** :Bangkok : Book Athens ltd.,part
- [3] ชาลยูทธ จันทรเด. **แผนที่เส้นทางในความควบคุมสำนักทางหลวงที่ 4**. พิษณุโลก : สำนักงานทางหลวงที่ 4
- [4] จัฑฑุฒิ พิษผล, พิชิต ตันติคุณานนท์. **คู่มือเรียน Visual Basic 6 ครั้งที่ 1**. กรุงเทพฯ : บ. โปรวิชั่น จำกัด. 2542
- [5] ประทีป ศรีธณโสภาส. **โครงสร้างข้อมูลและอัลกอริทึม**. พิษณุโลก : ภาควิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร.



## ประวัติผู้เขียน

นายวุฒิภูมิ อคนิษฐ์

เกิดเมื่อ วันที่ 20 มกราคม 2524 ที่จังหวัดพิจิตร

ภูมิลำเนา 56/1 หมู่ 11 ตำบลบางไผ่ อำเภอบางมูลนาก จังหวัดพิจิตร

จบการศึกษาชั้นประถมศึกษาจาก โรงเรียนครุฑบัณฑิตพิทยา จังหวัดพิจิตร

จบการศึกษาชั้นมัธยมศึกษาจาก โรงเรียนตะพานหิน จังหวัดพิจิตร

ปัจจุบันกำลังศึกษาระดับปริญญาตรีอยู่ชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

นายจักรชัย ใจเข้มแข็ง

เกิดเมื่อ วันที่ 6 พฤศจิกายน 2522 ที่จังหวัดมหาสารคาม

ภูมิลำเนา 492/28 ถนนศรีสวัสดิ์ดำเนิน ตำบลตลาด อำเภอเมือง จังหวัดมหาสารคาม

จบการศึกษาชั้นประถมศึกษาจาก โรงเรียนสาริตสถานบันราชภัฏมหาสารคาม จังหวัดมหาสารคาม

จบการศึกษาชั้นมัธยมศึกษาจาก โรงเรียนสาริตคณะศึกษาศาสตร์ มหาวิทยาลัยขอนแก่น  
จังหวัดขอนแก่น

ปัจจุบันกำลังศึกษาระดับปริญญาตรีอยู่ชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์  
มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก