บรรณานุกรม

1. องค์การขนส่งมวลชนกรุงเทพฯ, รายงานประจำปีองค์การขนส่งมวลชนกรุงเทพฯ 2534
2. Kenneth A.Ross, Charles R.B.Wright Discrete Mathematics : Prentice Hall International, Inc., 1988
3. Robin J.Wilson Introduction to Graph Theory : A Subsidiary of Harcourt Brace Jovanovich Publishers, 1970
4. Ralph P.Grimaldi Discrete and Combinatorial Mathematics An Applied Introduction : Addison-Wesley Publishing Company, 1992
5. James A.Mchugh Algorithmic Graph Theory : Prentice-Hall International, Inc., 1990
6. RichardJohnson BaughDiscrete Mathematics : Macmillan Publishing Company, 1992
7. Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest. Introduction to Algorithms : McGraw-Hill, 1990
8. อภิศักดิ์ โสมอินทร์ แผนที่และการแปลความหมายจากแผนที่ : สำนักพิมพ์ไทยวัฒนาพานิช, 2525
9. พินิจ ภาวกุล พันเอก การอ่านและการใช้แผนที่ : สำนักพิมพ์อักษรเจริญทัศน์, 2525

ภาคผนวก

ภาคผนวก ก

โครงสร้างของแฟ้มข้อมูล

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลรถโดยสารประจำทางแต่ละสาย (BUSKEY.DAT)

ลักษณะโครงสร้าง

int busn	สายรถโดยสารประจำทาง
int busk	รหัสรถโดยสารประจำทาง
int nodef	จุดสถานีต้นทาง
int nodet	จุดสถานีปลายทาง
char bus_name[15]	รายละเอียดของรถโดยสารประจำทาง

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนสายรถโดยสารประจำทางที่มีอยู่

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

ลักษณะโครงสร้าง

int nodef	จุดต้นช่วงถนน
int nodet	จุดปลายช่วงถนน
char typemap	ประเภทถนน
int busn	สายรถโดยสารประจำทาง

รายละเอียด

ประเภทถนน (typemap) ได้แก่

r แทนถนน

h แทนถนนทางด่วน

สายรถโดยสารประจำทาง (busn) เป็นสายรถประจำทางที่ผ่านจุดต้นช่วงถนนไปจุดปลายช่วงถนนซึ่งถ้ามีหลายสาย จะบันทึกเรียงต่อกันไป โดยเว้นช่องว่างแต่ละสาย 1 ช่อง

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลตำแหน่งของจุดที่เกี่ยวกับเส้นทางเดินรถ (POSITION.DAT)

ลักษณะโครงสร้าง

int fnode	จุดที่เกี่ยวกับเส้นทางเดินรถ
int fposx	ตำแหน่งแนวนอน (แกน X) หยาบ
int fposxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int fposy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int fposyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนจุดที่มีอยู่

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลตำแหน่งของจุดขยาย (POSACR.DAT)

ลักษณะโครงสร้าง

int fnode	จุดขยาย
int fposx	ตำแหน่งแนวนอน (แกน X) หยาบ
int fposxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int fposy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int fposyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนจุดที่มีอยู่

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลส่วนโค้งของถนน (MAPR.DAT)

ลักษณะโครงสร้าง

int nodef	จุดที่หนึ่ง
int nodet	จุดที่สอง
int pos_n	จุดขยาย

รายละเอียด

จุดขยาย (pos_n) เป็นจุดขยายในช่วงถนนจุดที่หนึ่ง (nodef) กับจุดที่สอง (nodet) ซึ่งถ้ามีหลายจุดขยาย (pos_n) จะบันทึกเรียงต่อกันไป โดยเว้นช่องว่างแต่ละสาย 1 ช่อง

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลเส้นประกอบแผนที่ (MAPD.DAT)

ลักษณะโครงสร้าง

char typemap	ประเภทของลักษณะส่วนประกอบของแผนที่
int fposx	ตำแหน่งต้นแนวนอน (แกน X) หยาบ
int fposxd	ตำแหน่งต้นแนวนอน (แกน X) ละเอียด
int fposy	ตำแหน่งต้นแนวตั้ง (แกน Y) หยาบ
int fposyd	ตำแหน่งต้นแนวตั้ง (แกน Y) ละเอียด
int tposx	ตำแหน่งปลายแนวนอน (แกน X) หยาบ
int tposxd	ตำแหน่งปลายแนวนอน (แกน X) ละเอียด
int tposy	ตำแหน่งปลายแนวตั้ง (แกน Y) หยาบ
int tposyd	ตำแหน่งปลายแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

ประเภทของลักษณะส่วนประกอบของแผนที่ (typemap) ได้แก่

R แทนถนนที่ไม่มีสายรถประจำทางผ่าน

V แทนส่วนประกอบของแม่น้ำ

<u>ชื่อแฟ้มข้อมูล</u> แฟ้มข้อมูลสถานที่สำคัญ (IMPORT.DAT)

ลักษณะโครงสร้าง

char name_s[15]	ชื่อสถานที่สำคัญ
int posx	ตำแหน่งแนวนอน (แกน X) หยาบ
int posxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int posy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int posyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด
char fname_s[12]	ชื่อแฟ้มข้อมูลที่ใช้เก็บภาพสัญญลักษณ์ (FILENAME.ICN)

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนสถานที่สำคัญ

ในกรณีที่ไม่มีภาพสัญญลักษณ์จะไม่มีชื่อแฟ้มข้อมูลที่ใช้เก็บภาพสัญญลักษณ์ (fname_s)

<u>ชื่อแฟ้มข้อมูล</u> แฟ้มข้อมูลโรงแรม (HOTEL.DAT)

ลักษณะโครงสร้าง

char name_s[15]	ชื่อโรงแรม
int posx	ตำแหน่งแนวนอน (แกน X) หยาบ
int posxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int posy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int posyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนโรงแรม

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลโรงพยาบาล (HOSPITAL.DAT)

ลักษณะโครงสร้าง

char name_s[15]	ชื่อโรงพยาบาล
int posx	ตำแหน่งแนวนอน (แกน X) หยาบ
int posxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int posy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int posyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนโรงพยาบาล

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลสถานีตำรวจ (POLICE.DAT)

ลักษณะโครงสร้าง

char name_s[15]	ชื่อสถานีตำรวจ
int posx	ต าแหน่งแนวนอน (แกน X) หยาบ
int posxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int posy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int posyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนสถานีตำรวจ

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลห้างสรรพสินค้า (STORE.DAT)

ลักษณะโครงสร้าง

char name_s[15]	ชื่อห้างสรรพสินค้า
int posx	ตำแหน่งแนวนอน (แกน X) หยาบ
int posxd	ตำแหน่งแนวนอน (แกน X) ละเอียด
int posy	ตำแหน่งแนวตั้ง (แกน Y) หยาบ
int posyd	ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนห้างสรรพสินค้า

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลภาพสัญญลักษณ์ (FILENAME.ICN)

ลักษณะโครงสร้าง

int color_icon	รหัสสีที่ใช้เป็นภาพสัญญลักษณ์
int icon_imp[15]	รหัสของภาพสัญญลักษณ์

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลรถโดยสารประจำทางสายใหม่ (BUSNAME.DAT)

ลักษณะโครงสร้าง

char tbus	ประเภทสายรถโดยสารประจำทางสายใหม่
int busn	หมายเลขสายรถโดยสารประจำทางสายใหม่
char name_s[15]	รายละเอียดของสายรถ

รายละเอียด

ประเภทสายรถโดยสารประจำทางสายใหม่ (tbus) ได้แก่

""	เป็นรถโดยสารประจำทางธรรมดา
"A"	เป็นรถโดยสารประจำทางปรับอากาศ
"B"	เป็นรถโดยสารประจำทางปรับอากาศพิเศษ

รายละเอียดของสายรถ (name_s) เช่น ป้ายแดง ป้ายน้ำเงิน

ชื่อแฟ้มข้อมูล แฟ้มข้อมูลเด็กผู้ชาย (BOY.LOG)

ลักษณะโครงสร้าง

char data	ข้อมูล

รายละเอียด

มีจำนวนระเบียนเท่ากับจำนวนจุดของภาพเด็กผู้ชายในแนวตั้งคือ 180 ระเบียน

แต่ละระเบียนเท่ากับจำนวนตัวอักษรเท่ากับจำนวนจุดของภาพเด็ผู้ชายในแนวนอนคือ 67 ตัวอักษร

โดยแต่ละตัวอักษรจะแทนสีของจุดหนึ่งจุด ซึ่งได้แก่

1 = สีน้ำเงินเข้ม (BLUE)	2 = สีเหลือง (YELLOW)
4 = สีแดง (RED)	5 = สีขาว (WHITE)
6 = สีน้ำตาล (BROWN)	8 = สีเทา (DARKGRAY)
9 = สีน้ำเงินอ่อน (LIGHTBLUE)	0 = สีขาว (WHITE)

<u>ชื่อแฟ้มข้อมูล</u> แฟ้มข้อมูลชื่อโปรแกรมสำเร็จรูปและชื่อผู้พัฒนา (NAME.LOG)

ลักษณะโครงสร้าง

unsigned int data ข้อมูล

รายละเอียด

ตัวอักษรที่ใช้แสดงเป็นชื่อโปรแกรมสำเร็จรูปและชื่อผู้พัฒนา 1 ตัวอักษร จะมีขนาด 16 * 32 จุด

ดังนั้นข้อมูล 1 ตัว จะแทนข้อมูลของจุดในแนวนอนหนึ่งแถว คือ 16 จุด

โดยจะถูกเปลี่ยนเป็นเลขฐานสอง จึงจะสื่อในการแสดง โดยจะเป็นจุดเรียงกัน โดย 0 จะเป็นสีพื้น 1 จะเป็นจุดสีขาว

ตัวอย่างเช่น

7792 = 1E70 H = 0001 1110 0111 0000 B

จะเป็น

สีพื้น สีพื้น สีพื้น สีขาว สีขาว สีขาว สีขาว สีพื้น ...

ภาคผนวก ช

โครงสร้างของข้อมูลที่อยู่ในหน่วยความจำ

ชื่อโครงสร้าง ตารางตำแหน่งของจุดที่เกี่ยวกับเส้นทางเดินรถ (table_pos)

โปรแกรมที่ใช้งาน 1. โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

2. โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

3. โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

4. โปรแกรมปรับปรุงข้อมูลสถานที่และสร้างพื้นที่เพิ่ม

อ่านข้อมูลจาก แฟ้มข้อมูลตำแหน่งของจุดที่เกี่ยวกับเส้นทางเดินรถ (POSITION.DAT)

การใช้งาน 1. ใช้บอกตำแหน่งของจุดที่เกี่ยวกับเส้นทางเดินรถ

2. ใช้เพิ่มเติมจุดใหม่ จากโปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

3. ใช้เคลื่อนย้ายจุด จากโปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

ลักษณะโครงสร้าง

```
typedef struct {
        int posx;       ตำแหน่งแนวนอน (แกน X) หยาบ
        int posxd;      ตำแหน่งแนวนอน (แกน X) ละเอียด
        int posy;       ตำแหน่งแนวตั้ง (แกน Y) หยาบ
        int posyd;      ตำแหน่งแนวตั้ง (แกน Y) ละเอียด
} POS
```

รายละเอียด

จุดที่เกี่ยวกับเส้นทางเดินรถใดที่ไม่มีการใช้จะมีค่าตำแหน่งแนวนอนหยาบ (posx) เป็น -1

ตัวอย่างเช่น

จุดที่	posx	posxd	posy	posyd
1	10	3	20	4
2	30	5	23	5

จุด 1 จะอยู่ที่ตำแหน่ง (10*10+3, 20*10+4) = (103, 204)

จุด 2 จะอยู่ที่ตำแหน่ง (30*10+5, 23*10+5) = (305, 235)

<u>ชื่อโครงสร้าง</u> ตารางตำแหน่งของจุดขยาย (table_posacr)

โปรแกรมที่ใช้งาน 1. โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

2. โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

3. โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

4. โปรแกรมปรับปรุงข้อมูลสถานที่และสร้างพื้นที่เพิ่ม

อ่านข้อมูลจาก แฟ้มข้อมูลตำแหน่งของจุดขยาย (POSACR.DAT)

การใช้งาน 1. ใช้บอกตำแหน่งของจุดขยาย

2. ใช้เพิ่มเติม, ลบ หรือเคลื่อนย้ายจุด จากโปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

ลักษณะโครงสร้าง

```
typedef struct {
        int posx;       ตำแหน่งแนวนอน (แกน X) หยาบ
        int posxd;      ตำแหน่งแนวนอน (แกน X) ละเอียด
        int posy;       ตำแหน่งแนวตั้ง (แกน Y) หยาบ
        int posyd;      ตำแหน่งแนวตั้ง (แกน Y) ละเอียด
} POS
```

ตัวอย่างเช่น

จุดที่	pcsx	posxd	posy	posyd
1	18	3	21	4
2	25	4	22	5

จุดขยาย 1 จะอยู่ที่ตำแหน่ง (18*10+3, 21*10+4) = (183, 214)

จุดขยาย 2 จะอยู่ที่ตำแหน่ง (25*10+4, 22*10+5) = (254, 225)

<u>ชื่อโครงสร้าง</u> ตารางเส้นถนนที่รถโดยสารผ่าน (mapline)

โปรแกรมที่ใช้งาน 1. โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

2. โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

3. โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

4. โปรแกรมปรับปรุงข้อมูลสถานที่และสร้างพื้นที่เพิ่ม

อ่านข้อมูลจาก 1. แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

2. แฟ้มข้อมูลส่วนโค้งของถนน (MAPR.DAT)

การใช้งาน 1. ใช้วาดเส้นถนนที่รถโดยสารผ่าน

2. ใช้เพิ่มเติม หรือลบข้อมูลสายรถสายสุดท้ายที่ใช้งานออกไป จากโปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_f;      จุดต้นช่วงถนน
        int pos_t;      จุดปลายช่วงถนน
        char type_m;    ประเภทถนน
        int link_m;     ตัวชี้ส่วนโค้ง
} MAPLINE
```

แต่ในโปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทางจะมีระยะเพิ่มด้วย เพื่อใช้ในการหาระยะทางสั้นที่สุด เป็น

```
typedef struct {
        int pos_f;      จุดต้นช่วงถนน
        int pos_t;      จุดปลายช่วงถนน
        float dis_l;    ระยะทาง
        char type_m;    ประเภทถนน
        int link_m;     ตัวชี้ส่วนโค้ง
} MAPLINE
```

รายละเอียด

ระยะทาง (dis_l) จะเป็นจากจุดต้นถนนถึงจุดปลายถนน ซึ่งจะคิดตามส่วนขยายที่ทำให้เกิดความโค้งด้วย

ประเภทถนน (type_m) จะแบ่งเป็น

r แทนถนน	ซึ่งจะเป็นสีขาว (WHITE)
h แทนทางด่วน	ซึ่งจะเป็นสีเขียวน้ำเงิน (CYAN)

ตัวชี้ส่วนโค้ง (link_m) จะมีดังนี้

ถ้าเป็น -1 คือ ไม่มีส่วนโค้ง จะเป็นเส้นตรงจากจุดต้นถนนถึงจุดปลายถนน

ถ้าเป็นตัวเลขที่มากกว่าหรือเท่ากับศูนย์ จะเป็นตัวชี้ชุดของสายรายการของจุดที่จะเชื่อมต่อกันเป็นถนนที่ไม่เป็นเส้นตรง

ตัวอย่างเช่น

ถ้า mapline เป็น 1, 2, r, -1 จะเป็นถนนเส้นตรงจากตำแหน่ง (103, 204) ไปยังตำแหน่ง (305, 235)

ถ้า mapline เป็น 1, 2, r, 0 โดยที่ mapacr[0]→1→2→ NULL จะเป็นถนนเส้นตรงจากตำแหน่ง (103, 204) ไปยังตำแหน่ง (183, 214) และจะต่อไปยังตำแหน่ง (254, 225) และต่อไปยังตำแหน่ง (305, 235) จึงได้รูปหลายเหลี่ยม

ถ้าถนนมีความโค้ง เราสามารถใช้สายรายการให้ยาว เพื่อให้รูปหลายเหลี่ยมมีส่วนใกล้เคียงส่วนโค้งมากยิ่งขึ้น

<u>ชื่อโครงสร้าง</u>	ชุดสายรายการจุดขยาย (mapacr)
โปรแกรมที่ใช้งาน	1. โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง
	2. โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย
	3. โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ
	4. โปรแกรมปรับปรุงข้อมูลสถานที่และสร้างพื้นที่เพิ่ม
อ่านข้อมูลจาก	แฟ้มข้อมูลส่วนโค้งของถนน (MAPR.DAT)
การใช้งาน	1. ใช้บอกทางเดินของเส้นถนนที่รถโดยสารผ่านที่มีจุดขยาย
	2. ใช้เพิ่มเติม หรือลบข้อมูลจุดขยาย จากโปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

ลักษณะโครงสร้าง

```
typedef struct mapacrnode {
        int  pos;
        struct mapacrnode *next;
} MAPACRNODE
```

ตัวอย่างเช่น

เมื่ออ่านข้อมูลจากแฟ้มข้อมูลระเบียนแรกเป็น 1 2 1 2

1 ตัวแรก แสดงถึง จุดต้นช่วงที่จะขยาย

2 ตัวแรก แสดงถึง จุดปลายช่วงที่จะขยาย

1 ตัวตัวที่สอง แสดงถึง จุดขยายต่อจากจุดต้นช่วงหรือจุดปลายช่วงขึ้นอยู่กับว่าตัวใดน้อยกว่า

2 ตัวตัวที่สอง แสดงถึง จุดขยายต่อจากจุดขยาย 1 และเป็นตัวสุดท้ายของการขยาย ดังนั้นจุดต่อไปคือจุดต้นช่วงหรือจุดปลายช่วง ขึ้นอยู่กับว่าตัวใดมากกว่า

เนื่องจากเป็นข้อมูลระเบียนแรก จึงอยู่เป็นสายแรกของชุด คือชุดที่ 0 คือ

mapacr[0]⟶1⟶2⟶NULL

หลังจากนั้นนำ จุดต้นของช่วงและจุดปลายของช่วง ไปค้นหาในตารางเส้นถนนที่รถโดยสารผ่าน (mapline) ว่าแถวใดที่เป็นช่วงถนนเดียวกัน ทั้งจุดต้นของช่วงเท่ากับจุดต้นช่วงถนน (pos_f) จุดปลายของช่วงเท่ากับจุดปลายช่วงถนน (pos_t) และ จุดต้นของช่วงเท่ากับจุดปลายช่วงถนน (pos_t) จุดปลายของช่วงเท่ากับจุดต้นช่วงถนน (pos_f) จะนำชุดที่ไปใส่ในตัวชี้ส่วนโค้ง (link_m) ของแถว

ดังนั้น mapline ที่ pos_f = 1 และ pos_t = 2 จะมีค่า link_m = 0 นั่นคือ 1, 2, r, 0 และในทำนองเดียวกัน mapline ที่ pos_f = 2 และ pos_t = 1 จะมีค่า link_m = 0 นั่นคือ 2, 1, r, 0 เช่นกัน

<u>ชื่อโครงสร้าง</u> ตารางเส้นประกอบแผนที่ (mapcomp)

โปรแกรมที่ใช้งาน 1. โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

2. โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

3. โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

4. โปรแกรมปรับปรุงข้อมูลสถานที่และสร้างพื้นที่เพิ่ม

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นประกอบแผนที่ (MAPD.DAT)

การใช้งาน 1. ใช้วาดเส้นประกอบแผนที่

2. ใช้เพิ่มเติม หรือลบข้อมูลเส้นประกอบแผนที่ จากโปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_fx;      ตำแหน่งต้นแนวนอน
        int pos_fy;      ตำแหน่งต้นแนวตั้ง
        int pos_tx;      ตำแหน่งปลายแนวนอน
        int pos_ty;      ตำแหน่งปลายแนวตั้ง
        char type_m;     ประเภทเส้น
} MAPCOMP
```

รายละเอียด

ตำแหน่งต้นแนวนอน (แกน X) คือ pos_fx = ตำแหน่งต้นแนวนอน (แกน X) หยาบ * 10 + ตำแหน่งต้นแนวนอน (แกน X) ละเอียด

ตำแหน่งต้นแนวตั้ง (แกน Y) คือ pos_fy = ตำแหน่งต้นแนวตั้ง (แกน Y) หยาบ * 10 + ตำแหน่งต้นแนวตั้ง (แกน Y) ละเอียด

ตำแหน่งปลายแนวนอน (แกน X) คือ pos_tx = ตำแหน่งปลายแนวนอน (แกน X) หยาบ * 10 + ตำแหน่งปลายแนวนอน (แกน X) ละเอียด

ตำแหน่งปลายแนวตั้ง (แกน Y) คือ pos_ty = ตำแหน่งปลายแนวตั้ง (แกน Y) หยาบ * 10 + ตำแหน่งปลายแนวตั้ง (แกน Y) ละเอียด

ประเภทของเส้น (type_m) จะแบ่งเป็น

R แทนถนนที่ไม่มีสายรถประจำทางผ่าน จะมีสีเทา (LIGHTGRAY)

V แทนส่วนประกอบของแม่น้ำ จะมีสีเขียวแก่ (GREEN)

X แทนส่วนว่าเส้นประกอบนี้ได้ถูกลบออกแล้ว

เมื่อต้องการวาดแผนที่จะใช้ลากเส้นจาก (pos_fx, pos_fy) ไปยัง (pos_tx, pos_ty) โดยใช้ type_m เป็นตัวบ่งบอกสีของเส้น

<u>ชื่อโครงสร้าง</u> ตารางสัญญลักษณ์สถานที่สำคัญ (impicon)

โปรแกรมที่ใช้งาน 1. โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

2. โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

3. โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

อ่านข้อมูลจาก 1. แฟ้มข้อมูลสถานที่สำคัญ (IMPORT.DAT)

2. แฟ้มข้อมูลสัญญลักษณ์สถานที่สำคัญ (FILENAME.ICN)

การใช้งาน ใช้วาดสัญญลักษณ์สถานที่สำคัญประกอบแผนที่

ลักษณะโครงสร้าง

```
typedef struct {
        char name_s[26];    ชื่อสถานที่
        int pos_x;          ตำแหน่งแนวนอน
        int pos_y;          ตำแหน่งแนวตั้ง
        int color_icon;     รหัสสีของภาพสัญญลักษณ์
        unsigned int imp_icon[15];
                            ชุดข้อมูลภาพสัญญลักษณ์
} IMPICON
```

รายละเอียด

ชื่อสถานที่ (name_s) มีความยาวไม่เกิน 25 ตัวอักษร

ตำแหน่งแนวนอน (แกน X) คือ pos_x = ตำแหน่งแนวนอน (แกน X) หยาบ * 10
+ ตำแหน่งแนวนอน (แกน X) ละเอียด

ตำแหน่งแนวตั้ง (แกน Y) คือ pos_y = ตำแหน่งแนวตั้ง (แกน Y) หยาบ * 10
+ ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

รหัสสีของภาพสัญญลักษณ์ ได้แก่

1 = สีน้ำเงินเข้ม (BLUE)	2 = สีเขียวแก่ (GREEN)
3 = สีเขียวน้ำเงิน (CYAN)	4 = สีแดง (RED)
5 = สีม่วง (MAGENTA)	6 = สีน้ำตาล (BROWN)
9 = สีน้ำเงินอ่อน (LIGHTBLUE)	10 = สีเขียวอ่อน (LIGHTGREEN)
11 = สีฟ้าอ่อน (LIGHTCYAN)	12 = สีส้ม (LIGHTRED)
13 = สีบานเย็น (LIGHTMAGENTA)	14 = สีเหลือง (YELLOW)

นำชื่อแฟ้มข้อมูลที่ใช้เก็บภาพสัญญลักษณ์ (FILENAME.ICN) ไปเปิดเพื่อหาข้อมูลของรหัสสีของภาพสัญญลักษณ์และข้อมูลของภาพสัญญลักษณ์

ข้อมูลภาพสัญญลักษณ์ในตัวที่ n จะเก็บข้อมูลภาพสัญญลักษณ์แถวที่ n

ข้อมูลแต่ละตัวจะถูกเปลี่ยนเป็นเลขฐานสอง จึงจะสื่อในการแสดงโดยจะเป็นจุดเรียงกัน โดย 0 จะเป็นสีขาว 1 จะเป็นสีตามรหัสสีของภาพสัญญลักษณ์

แนวนอนใช้ 15 ตัว และแนวตั้งมี 15 ตัว ดังนั้นภาพสัญญลักษณ์ของสถานที่สำคัญจะมีขนาด 15 * 15 จุด

ตัวอย่างเช่น

imp_icon = 7792, 2032, ...

สีภาพสัญญลักษณ์เป็นสีเหลือง

7792 = 1E70 H = 001 1110 0111 0000 B

2032 = 07F0 H = 000 0111 1111 0000 B

ภาพสัญญลักษณ์แถวแรก จะแสดงเป็น

ขาว ขาว เหลือง เหลือง เหลือง เหลือง ขาว ขาว ...

<u>ชื่อโครงสร้าง</u> ตัวแปรเก็บตำแหน่งต้นทาง (fstation)

ตัวแปรเก็บตำแหน่งปลายทาง (tstation)

ตัวแปรเก็บตำแหน่งที่ไม่ต้องการผ่าน (nstation)

ตัวแปรเก็บตำแหน่งที่ต้องการผ่าน (pstation)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ผู้ใช้ป้อนตำแหน่งที่ต้องการ

ลักษณะโครงสร้าง

```
typedef struct {
        int road_f;      จุดต้นช่วงถนนที่ใกล้จุดที่เลือก
        char troad_f;    ประเภทจุดต้นช่วงถนน
        int road_t;      จุดปลายช่วงถนนที่ใกล้จุดที่เลือก
        char troad_t;    ประเภทจุดปลายช่วงถนน
        int row_m;       แถวตารางเส้นถนนที่รถโดยสารผ่าน
        int pos_x;       ตำแหน่งแนวนอน
        int pos_y;       ตำแหน่งแนวตั้ง
} ROAD
```

รายละเอียด

ในกรณีตำแหน่งที่ผู้ใช้เลือก

1. เป็นจุดที่เกี่ยวกับเส้นทางเดินรถ

จุดต้นเส้นที่ใกล้จุดที่เลือก (road_f) = จุดที่เกี่ยวกับเส้นทางเดินรถ

ประเภทจุดต้นเส้น (troad_f) = 'R'

จุดปลายเส้นที่ใกล้จุดที่เลือก (road_t) = จุดที่เกี่ยวกับเส้นทางเดินรถ

ประเภทจุดปลายเส้น (troad_t) = 'R'

แถวตารางเส้นถนนที่รถโดยสารผ่าน (row_m) = -1

2. ไม่เป็นจุดที่เกี่ยวกับเส้นทางเดินรถ โปรแกรมจะคำนวณหาตำแหน่งบนเส้นทางที่ใกล้ที่สุด เป็นจุดขึ้นหรือลงรถแทน ซึ่งเส้นนี้อาจจะเป็นระหว่างจุดที่เกี่ยวกับเส้นทางเดินรถกับจุดที่เกี่ยวกับเส้นทางเดินรถ, จุดที่เกี่ยวกับเส้นทางเดินรถกับจุดขยาย หรือจุดขยายกับจุดขยาย

ประเภทจุดต้นเส้น (troad_f) คือ

'R' ถ้าจุดต้นเส้นเป็นจุดที่เกี่ยวกับเส้นทางเดินรถ

'A' ถ้าจุดต้นเส้นเป็นจุดขยาย

ประเภทจุดปลายเส้น (troad_t) คือ

'R' ถ้าจุดปลายเส้นเป็นจุดที่เกี่ยวกับเส้นทางเดินรถ

'A' ถ้าจุดปลายเส้นเป็นจุดขยาย

แถวตารางเส้นถนนที่รถโดยสารผ่าน (row_m) คือ

แถวของตารางเส้นถนนที่รถโดยสารผ่านที่จะเป็นจุดที่ใช้

3. เป็นการเลือกโดยชื่อของสถานที่ จะนำตำแหน่งของสถานที่มาเป็นตำแหน่งที่ต้องการแทน ซึ่งอาจเป็นจุดที่เกี่ยวกับเส้นทางเดินรถหรือไม่เป็นจุดที่เกี่ยวกับเส้นทางเดินรถแล้วแต่กรณีไป

ตำแหน่งแนวนอน (pos_x) จะเป็นตำแหน่งของจุดที่ใช้บนเส้น

ตำแหน่งแนวตั้ง (pos_y) จะเป็นตำแหน่งของจุดที่ใช้บนเส้น

<u>ชื่อโครงสร้าง</u> ตารางทำงานหาเส้นทางสั้นที่สุด (wkshort)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้ในการหาเส้นทางที่สั้นที่สุด

ลักษณะโครงสร้าง

```
typedef struct {
        float dis_l;        ระยะที่ห่างจากจุดปลายทาง
        int pos_t;          จุดที่จะไปต่อไป
        int flag_s;         รหัสการเข้าทำ
} WKSHORT
```

รายละเอียด

จุดที่จะไปต่อไป (pos_t) เป็นตัวบอกเส้นเดินที่จะได้ระยะทางสั้นที่สุด หลังผ่านกระบวนการหาเส้นทางสั้นที่สุดแล้ว โดยจะเริ่มต้นเดินจากจุดเริ่มต้นจนกระทั่งพบจุดปลายที่ต้องการ

รหัสการเข้าทำ (flag_s) เป็นตัวบอกว่าจุดใดที่เข้าไปในกระบวนการหาเส้นทางสั้นที่สุดแล้ว

ชื่อโครงสร้าง ตารางเส้นถนนที่รถโดยสารผ่านที่เรียงลำดับตามจุดปลายช่วงถนน (maplinerev)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้หาแถวของตารางเส้นถนนที่รถโดยสารผ่าน (mapline)

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_f;      จุดต้นช่วงถนน
        int pos_t;      จุดปลายช่วงถนน
        int row_m;      แถวตารางเส้นถนนที่รถโดยสารผ่าน
} MAPLINEREV
```

รายละเอียด

จุดต้นช่วงถนน (pos_f) และจุดปลายช่วงถนน (pos_t) จะมีเหมือนกับตารางเส้นถนนที่รถโดยสารผ่าน (mapline) แต่ตารางเส้นถนนที่รถโดยสารผ่าน (mapline) จะเรียงตามลำดับตามจุดต้นช่วงถนน (pos_f) ส่วนตารางเส้นถนนที่รถโดยสารผ่านที่เรียงลำดับตามจุดปลายช่วงถนน (maplinerev) จะเรียงตามลำดับตามจุดปลายช่วงถนน (pos_t)

แถวตารางเส้นถนนที่รถโดยสารผ่าน (row_m) เป็นตัวชี้ว่าเมื่อจุดต้นช่วงถนน (pos_f) และจุดปลายช่วงถนน (pos_t) เท่ากันแล้ว จะมีแถวในตารางเส้นถนนที่รถโดยสารผ่าน (mapline) เป็นเท่าไร

ซึ่งใช้ในการช่วยหาแถวของตารางเส้นถนนที่รถโดยสารผ่าน (mapline) ที่มีจุดปลายช่วงถนนเท่ากับที่กำหนด ได้รวดเร็วกว่าค้นหาทั้งตารางตารางเส้นถนนที่รถโดยสารผ่าน (mapline)

<u>ชื่อโครงสร้าง</u> ตารางเส้นทางที่จะแนะนำ (busroute)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้ในการแสดงเส้นทางที่จะแนะนำ

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_f;          จุดต้นช่วงถนน
        int pos_t;          จุดปลายช่วงถนน
        int row_m;          แถวตารางเส้นถนนที่รถโดยสารผ่าน
        struct busnode *head_buslist;
                            สายรายการของสายรถที่ผ่านจากจุดต้นช่วงถนนไปยังจุด
                            ปลายช่วงถนน
} BUSROUTE
```

รายละเอียด

ตารางจะเรียงลำดับจากจุดเริ่มต้นไปจนถึงจุดปลายทาง

และสายรายการของสายรถที่ผ่านจากจุดต้นช่วงถนนไปยังจุดปลายช่วงถนน จะเป็นตัวบอกสายรถโดยสารประจำทางที่ต้องใช้

แต่ละแถวของตารางอาจจะมีสายรถโดยสารประจำทางที่ผ่านซ้ำกับแถวก่อนหน้านี้ จึงใช้แนะนำจุดต่อรถโดยสารประจำทางไม่ได้

ตัวอย่างเช่น

busroute[0] = 1, 2, 1, →34→54→NULL

busroute[1] = 2, 5, 5, →34→36→47→NULL

busroute[2] = 5, 7, 10, →36→96→NULL

เส้นทางที่จะแนะนำคือ เริ่มต้นที่จุด 1 ต่อไปยังจุด 2 ต่อไปยังจุด 5 ต่อไปยังจุด 7 ซึ่งเป็นจุดปลายทาง

<u>ชื่อโครงสร้าง</u> ตารางเส้นทางที่จะแนะนำชั่วคราว (busroutetemp)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

การใช้งาน ใช้ในการใส่ข้อมูลสายรถเข้าไปในตารางเส้นทางที่จะแนะนำ (busroute)

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_f;          จุดต้นช่วงถนน
        int pos_t;          จุดปลายช่วงถนน
        int row_m;          แถวตารางเส้นถนนที่รถโดยสารผ่าน
        int row_b;          แถวตารางเส้นทางที่จะแนะนำ
        struct busnode *head_buslist;
                            สายรายการของสายรถที่ผ่านจากจุดต้นช่วงถนนไปยัง
                            จุดปลายช่วงถนน
} BUSROUTETEMP
```

รายละเอียด

ตารางเส้นทางที่จะแนะนำชั่วคราว (busroutetemp) จะเหมือนตารางเส้นทางที่จะแนะนำ (busroute) แต่ตารางเส้นทางที่จะแนะนำชั่วคราวจะเรียงลำดับตามจุดต้นช่วงถนน (pos_f) และจุดปลายช่วงถนน (pos_t) เพื่อใช้ในการอ่านข้อมูลสายรถโดยสารประจำทางที่ผ่านในช่วงถนนดังกล่าว ภายในรอบเดียวเพราะแฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT) จะเรียงตามลำดับจุดต้นช่วงถนน (pos_f) และจุดปลายช่วงถนน (pos_t) หลังจากพบช่วงที่ตรงกันก็จะนำมาใส่สายรายการของสายรถที่ผ่านจากจุดต้นช่วงถนนไปยังจุดปลายช่วงถนน เมื่ออ่านข้อมูลครบจะทำการเชื่อมข้อมูลสายรายการของสายรถที่ผ่านจากจุดต้นช่วงถนนไปยังจุดปลายกับตารางตารางเส้นทางที่จะแนะนำ (busroute) ซึ่งจะมีแถวตารางเส้นทางที่จะแนะนำ (row_b) เป็นตัวชี้

<u>ชื่อโครงสร้าง</u> ตารางจุดต่อรถที่จะแนะนำ (busconnect)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้ในการแนะนำสายรถและจุดต่อ

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_f;          จุดเริ่มขึ้นรถ
        int pos_t;          จุดลงรถ
        struct busflagnode *head_buslist;
                            สายรายการของสายรถที่ผ่านจากจุดเริ่มขึ้นรถไปยัง
                            จุดลงรถ
} BUSCONNECT
```

รายละเอียด

ตารางจะเรียงลำดับจากจุดเริ่มต้นไปจนถึงจุดปลายทาง

และสายรายการของสายรถที่ใช้จากจุดเริ่มขึ้นไปยังจุดลงรถ จะเป็นตัวบอกสายรถโดยสารประจำทางที่ต้องใช้

ตัวอย่างเช่น

busconnect[0] = 1, 5, →34

busconnect[1] = 5, 7, →36

จุดต่อรถที่จะแนะนำ คือ ขึ้นรถสาย 34 ที่จุด 1 และลงรถที่จุด 5 ขึ้นรถสาย 36 ต่อและลงรถที่จุด 7 จะถึงปลายทาง

ชื่อโครงสร้าง สายรายการรถโดยสารประจำทางที่ออกจากตำแหน่งต้นทาง (head_busposf)
โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง
อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)
การใช้งาน ใช้เก็บสายรถที่ออกจากตำแหน่งต้นทาง
ลักษณะโครงสร้าง

```
typedef struct busrowm {
        int bus_n;          สายรถที่ออกจากตำแหน่งต้นทาง
        int row_m;          แถวตารางเส้นถนนที่รถโดยสารผ่าน
        int flag_f;         รหัสการพบ
        struct busrowm *next;
} BUSROWM
```

รายละเอียด

แถวตารางเส้นถนนที่รถโดยสารผ่าน (row_m) เป็นแถวของช่วงถนนที่ออกจากตำแหน่งต้นทาง

ในสายรายการจะต้องไม่มีหน่วยที่ทั้งสายรถที่ออกจากตำแหน่งต้นทาง (bus_n) และแถวตารางเส้นถนนที่รถโดยสาร (row_m) เท่ากัน

ชื่อโครงสร้าง สายรายการรถโดยสารประจำทางที่ไปถึงตำแหน่งปลายทาง (head_buspost)
โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง
อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)
การใช้งาน ใช้เก็บสายรถที่ไปถึงตำแหน่งปลายทาง
ลักษณะโครงสร้าง

```
typedef struct busflagnode {
        int bus_n;          สายรถที่ไปถึงตำแหน่งปลายทาง
        int flag_f;         รหัสการพบ
        struct busflagnode *next;
} BUSFLAGNODE
```

รายละเอียด

ในสายรายการจะต้องไม่มีหน่วยที่ทั้งสายรถที่ไปถึงตำแหน่งปลายทาง (bus_n)

<u>ชื่อโครงสร้าง</u> ตารางเก็บสายรถ (busall)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้เก็บสายรถที่ผ่านในบางส่วน

ลักษณะโครงสร้าง

```
typedef struct {
        int row_m;              แถวตารางเส้นถนนที่รถโดยสารผ่าน
        struct busnode *head_buslist;
} BUSALL
typedef struct busnode {
        int bus_n;              สายรถที่ผ่าน
        struct busnode *next;
} BUSNODE
```

รายละเอียด

ตารางเก็บสายรถ จะไม่ได้เก็บข้อมูลทั้งหมดของแฟ้มข้อมูล จะเก็บเฉพาะบางส่วนขึ้นอยู่กับเงื่อนไขที่ต้องการ

<u>ชื่อโครงสร้าง</u> สายรายการแบบการต่อรถหนึ่งต่อ (head_buspass)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้เก็บข้อมูลแบบของการเดินทางโดยใช้การต่อรถหนึ่งต่อ

ลักษณะโครงสร้าง

```
typedef struct buspass {
        int frow_m;        แถวตารางเส้นถนนที่รถโดยสารผ่าน
        int bus_n;         สายรถ
        struct buspass *next;
} BUSPASS
```

รายละเอียด

แถวตารางเส้นถนนที่รถโดยสารผ่าน (frow_m) เป็นแถวที่เป็นช่วงถนนที่ออกจากตำแหน่งต้นทาง

สายรถ (bus_n) เป็นสายรถที่ออกตำแหน่งต้นทางและสามารถเดินทางไปถึงตำแหน่งปลายทางได้

ข้อมูลได้จากสายรายการรถโดยสารประจำทางที่ออกจากตำแหน่งต้นทาง (head_busposf) กับ สายรายการรถโดยสารประจำทางที่ไปถึงตำแหน่งปลายทาง (head_buspost) ที่มีสายรถ (bus_n) เท่ากัน

<u>ชื่อโครงสร้าง</u> สายรายการแบบการต่อรถสองต่อ (head_bus2con)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

การใช้งาน ใช้เก็บข้อมูลแบบของการเดินทางโดยใช้การต่อรถสองต่อ

ลักษณะโครงสร้าง

```
typedef struct bus2con {
        int bus_f;          สายรถที่ออกจากตำแหน่งต้นทาง
        int row1_m;         แถวตารางเส้นถนนที่รถโดยสารผ่าน 1
        int row2_m;         แถวตารางเส้นถนนที่รถโดยสารผ่าน 2
        int bus_t;          สายรถที่ไปถึงตำแหน่งปลายทาง
        struct bus2con *next;
} BUS2CON
```

รายละเอียด

สายรถที่ออกจากตำแหน่งต้นทาง (bus_f) จะมีจุดที่สามารถต่อรถสายรถที่ไปถึงตำแหน่งปลายทาง (bus_t)

แถวตารางเส้นถนนที่รถโดยสารผ่าน 1 (row1_m) เป็นแถวที่เป็นช่วงถนนที่ออกจากตำแหน่งต้นทาง

แถวตารางเส้นถนนที่รถโดยสารผ่าน 2 (row2_m) เป็นแถวที่เป็นช่วงถนนที่ออกจากตำแหน่งต้นทางอีกทาง ซึ่งถ้ามีทางเดียวจะเท่ากับแถวตารางเส้นถนนที่รถโดยสารผ่าน 1 (row1_m)

แถวที่เป็นเส้นทางออกจากตำแหน่งต้นทางอาจจะได้มากกว่า 2 ทาง และแถวที่เป็นเส้นทางออกจากตำแหน่งต้นทางนี้ ได้ข้อมูลมาจากสายรายการรถโดยสารประจำทางที่ผ่านตำแหน่งต้นทาง (head_busposf) ซึ่งได้กำจัดแถวที่ไม่มีโอกาสเดินถึงตำแหน่งปลายทางออกแล้ว

ชื่อโครงสร้าง สายรายการสายรถที่ต่อได้จากสายที่ออกจากตำแหน่งต้นทาง (head_bus2listf)

โปรแกรมที่ใช้งาน โปรแกรมแนะน

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

การใช้งาน ใช้เก็บแบบของสายรถที่ต่อได้จากสายที่ออกจากตำแหน่งต้นทาง เพื่อใช้ในการหาแบบของการเดินทางโดยใช้การต่อรถสามต่อต่อไป

ลักษณะโครงสร้าง

```
typedef struct bus2list {
        int bus_1;      สายรถที่ออกจากตำแหน่งต้นทาง
        int bus_2;      สายรถที่ต้องต่อ
        struct  bus2list *next;
} BUS2LIST
```

รายละเอียด

สายรถที่ออกจากตำแหน่งต้นทาง (bus_1) จะมีจุดที่สามารถต่อรถสายรถที่ต้องต่อ (bus_2) ได้

ชื่อโครงสร้าง สายรายการสายรถที่ไปต่อสายรถไปถึงตำแหน่งปลายทาง (head_bus2listt)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

การใช้งาน ใช้เก็บแบบของสายรถที่ไปต่อสายรถไปถึงตำแหน่งปลายทาง เพื่อใช้ในการหาแบบของการเดินทางโดยใช้การต่อรถสามต่อต่อไป

ลักษณะโครงสร้าง

```
typedef struct bus2list {
        int bus_1;          สายรถที่ไปถึงตำแหน่งปลายทาง
        int bus_2;          สายรถที่ต้องต่อ
        struct  bus2list *next;
} BUS2LIST
```

รายละเอียด

สายรถที่ต้องต่อ (bus_2) จะมีจุดที่สามารถต่อรถสายรถที่ไปถึงตำแหน่งปลายทาง (bus_1)

ชื่อโครงสร้าง สายรายการแบบการต่อรถสามต่อ (head_bus3con)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

การใช้งาน ใช้เก็บข้อมูลแบบของการเดินทางโดยใช้การต่อรถสามต่อ

ลักษณะโครงสร้าง

```
typedef struct bus3con {
        int row1_m;         แถวตารางเส้นถนนที่รถโดยสารผ่าน 1
        int row2_m;         แถวตารางเส้นถนนที่รถโดยสารผ่าน 2
        int buspos_f;       สายรถที่ออกจากตำแหน่งต้นทาง
        int bus_n;          สายรถที่ต้องต่อ
        int buspos_t;       สายรถที่ไปถึงตำแหน่งปลายทาง
        struct bus3con *next;
} BUS3CON
```

รายละเอียด

สายรถที่ออกจากตำแหน่งต้นทาง (buspos_f) สามารถไปต่อรถสายที่ต้องต่อ (bus_n) ที่จุดหนึ่ง และไปต่อรถสายรถที่ไปถึงตำแหน่งปลายทาง (buspos_t) อีกครั้งที่อีกจุด จะสามารถถึงตำแหน่งปลายทางได้

ข้อมูลได้จากสายรายการสายรถที่ต่อได้จากสายที่ออกจากตำแหน่งต้นทาง (head_bus2listf) กับสายรายการสายรถที่ไปต่อสายรถไปถึงตำแหน่งปลายทาง (head_bus2listt) ที่มีสายรถที่ต้องต่อ (bus_n) เท่ากัน

แถวตารางเส้นถนนที่รถโดยสารผ่าน 1 (row1_m) เป็นแถวที่เป็นช่วงถนนที่ออกจากตำแหน่งต้นทาง

แถวตารางเส้นถนนที่รถโดยสารผ่าน 2 (row2_m) เป็นแถวที่เป็นช่วงถนนที่ออกจากตำแหน่งต้นทางอีกทาง ซึ่งถ้ามีทางเดียวจะเท่ากับแถวตารางเส้นถนนที่รถโดยสารผ่าน 1 (row1_m)

แถวที่เป็นเส้นทางออกจากตำแหน่งต้นทางอาจจะมีได้มากกว่า 2 ทาง แต่แถวที่เป็นเส้นทางออกจากตำแหน่งต้นทางนี้ ได้ข้อมูลมาจากสายรายการรถโดยสารประจำทางที่ผ่านตำแหน่งต้นทาง (head_busposf) ซึ่งได้กำจัดแถวที่ไม่มีโอกาสเดินถึงตำแหน่งปลายทางออกแล้ว

ชื่อโครงสร้าง ตารางรหัสรถโดยสารประจำทางที่มีรายละเอียด (buskeyname)
โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง
อ่านข้อมูลจาก แฟ้มข้อมูลสายรถโดยสารประจำทางแต่ละสาย (BUSKEY.DAT)
การใช้งาน เพื่อใช้หารายละเอียดสายรถ
ลักษณะโครงสร้าง

```
typedef struct {
        int bus_k;          รหัสสายรถ
        char bus_name[16]; รายละเอียดสายรถ
} BUSKEYNAME
```

รายละเอียด

รหัสสายรถ (bus_k) เป็นคีย์หลัก (Primary key)

จะเก็บเฉพาะสายรถที่มีรายละเอียด

ชื่อโครงสร้าง ตารางรายชื่อสถานที่ (stname)

โปรแกรมที่ใช้งาน โปรแกรมแนะนำเส้นทางเดินรถโดยสารประจำทาง

อ่านข้อมูลจาก 1. แฟ้มข้อมูลสถานที่สำคัญ (IMPORT.DAT)

2. แฟ้มข้อมูลโรงแรม (HOTEL.DAT)

3. แฟ้มข้อมูลโรงพยาบาล (HOSPITAL.DAT)

4. แฟ้มข้อมูลสถานีตำรวจ (POLICE.DAT)

5. แฟ้มข้อมูลห้างสรรพสินค้า (STORE.DAT)

การใช้งาน ใช้หาตำแหน่งโดยชื่อของสถานที่หรือกระจายภาพสัญญลักษณ์ เพื่อหาตำแหน่ง

ลักษณะโครงสร้าง

```
typedef struct {
        char name_s[26];   ชื่อสถานที่
        int pos_x;         ตำแหน่งแนวนอน
        int pos_y;         ตำแหน่งแนวตั้ง
} STNAME
```

รายละเอียด

ชื่อสถานที่ (name_s) มีความยาวไม่เกิน 25 ตัวอักษร

ตำแหน่งแนวนอน (แกน X) คือ

pos_x = ตำแหน่งแนวนอน (แกน X) หยาบ * 10 +
ตำแหน่งแนวนอน (แกน X) ละเอียด

ตำแหน่งแนวตั้ง (แกน Y) คือ

pos_y = ตำแหน่งแนวตั้ง (แกน Y) หยาบ * 10 +
ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

ชื่อโครงสร้าง ตารางเส้นทางเดินรถ (busline)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

การใช้งาน 1. ใช้แสดงเส้นทางเดินของสายรถที่เลือกไว้

2. ใช้เพิ่มเติม หรือลบข้อมูลเส้นทางสายรถที่เลือกไว้

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_f;      จุดต้นช่วงถนน
        int pos_t;      จุดปลายช่วงถนน
        int direc;      ทิศทาง
        char type_m;    ประเภทถนน
        int link_m;     ตัวชี้ส่วนโค้ง
} BUSLINE
```

รายละเอียด

ทิศทาง (direc) จะมีค่าเป็น 3 ถ้าช่วงถนนนั้นมีรถสายที่เลือกผ่านทั้งขาไปและขากลับ

ทิศทาง (direc) จะมีค่าเป็น 1 ถ้าช่วงถนนนั้นมีรถสายที่เลือกผ่านเฉพาะขาไป

ทิศทาง (direc) จะมีค่าเป็น 0 ถ้าช่วงถนนนั้นไม่มีรถสายที่เลือกผ่านทั้งขาไปและขากลับ

ประเภทถนน (type_m) ได้แก่

ถนนธรรมดา จะเป็น 'r'

ถนนทางด่วน จะเป็น 'h'

ตัวชี้ส่วนโค้ง (link_m) จะเหมือนกับในตารางเส้นถนนที่รถโดยสารผ่าน (mapline) เพื่อสะดวกในการแสดงเส้นทาง

ชื่อโครงสร้าง ตารางชุดคำสั่งชั่วคราว (commandtran)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

อ่านข้อมูลจาก แฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

การใช้งาน ใช้เก็บคำสั่งที่ผู้ใช้เพิ่มเติม หรือลบข้อมูลเส้นทางเดิน เพื่อนำไปปรับปรุงกับแฟ้มข้อมูล

ลักษณะโครงสร้าง

```
typedef struct {
        char com_t;     ประเภทคำสั่ง
        int pos_f;      จุดต้นช่วงถนน
        int pos_t;      จุดปลายช่วงถนน
        char type_m;    ประเภทถนน
} COMMANDTRAN
```

รายละเอียด

ประเภทคำสั่ง (com_t) ได้แก่

การเพิ่มเติมเส้นทาง	ใช้ 'i'
การลบเส้นทาง	ใช้ 'd'

ประเภทถนน (type_m) ได้แก่

ถนนธรรมดา	ใช้ 'r'
ถนนทางด่วน	ใช้ 'h'

ชื่อโครงสร้าง สายรายการเส้นที่ลบได้ (head_remove)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

การใช้งาน ใช้เก็บข้อมูลเส้นที่ลบได้ คือได้ทำการลบข้อมูลของสายสุดท้ายที่ใช้ช่วงดังกล่าวออกไปแล้ว แล้วนำไปสร้างเป็นเส้นประกอบแผนที่แทน

ลักษณะโครงสร้าง

```
typedef struct noderemove {
        int row1_m;        แถวตารางเส้นถนนที่รถโดยสารผ่าน 1
        int row2_m;        แถวตารางเส้นถนนที่รถโดยสารผ่าน 2
        int direc;         ทิศทาง
        struct noderemove *next;
} NODEREMOVE
```

รายละเอียด

แถวตารางเส้นถนนที่รถโดยสารผ่าน 1 (row1_m) เป็นแถวที่ต้องการลบออกจากตารางเส้นถนนที่รถโดยสารผ่าน (mapline)

แถวตารางเส้นถนนที่รถโดยสารผ่าน 2 (row2_m) เป็นที่ต้องการลบออกจากตารางเส้นถนนที่รถโดยสารผ่าน (mapline) โดยอยู่ในช่วงถนนเดียวกันกับแถวตารางเส้นถนนที่รถโดยสารผ่าน1 (row1_m) แต่ในทิศทางสวนกลับ

ถ้ามีช่วงถนนหนึ่ง มีแถวที่จะลบทั้งขาไปและขากลับ ทิศทาง (direc) = 2 จะต้องนำเส้นนี้ไปเพิ่มในตารางเส้นประกอบแผนที่ (mapcomp) โดยมีประเภทเส้น (type_m) เป็น 'R'

ถ้ามีช่วงถนนหนึ่ง มีแถวที่จะลบทั้งขาไปอย่างเดียว ทิศทาง (direc) = 1 แถวตารางเส้นถนนที่รถโดยสารผ่าน1 (row1_m) = แถวตารางเส้นถนนที่รถโดยสารผ่าน2 (row2_m) จะต้องไปตรวจสอบก่อนว่าไม่มีการใช้เส้นนี้ในทิศทางสวนกลับ จึงจะต้องนำเส้นนี้ไปเพิ่มในตารางเส้นประกอบแผนที่ (mapcomp) โดยมีประเภทเส้น (type_m) เป็น 'R'

<u>ชื่อโครงสร้าง</u> ตารางคำสั่งชั่วคราวแทรกจุด (comroad)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

การใช้งาน ใช้เก็บคำสั่งที่ผู้ใช้แทรกจุด

ลักษณะโครงสร้าง

```
typedef struct {
        int road_f;      จุดต้นช่วงถนนที่ใกล้จุดที่แทรก
        char troad_f;    ประเภทจุดต้นช่วงถนน
        int road_t;      จุดปลายช่วงถนนที่ใกล้จุดที่แทรก
        char troad_t;    ประเภทจุดปลายช่วงถนน
        int row_m;       แถวตารางเส้นถนนที่รถโดยสารผ่าน
        int pos_o;       จุดเดิม
        int pos_n;       จุดใหม่
} COMROAD
```

รายละเอียด

ถ้าจุดใหม่ที่แทรก เป็นจุดขยายเดิม จุดเดิม (pos_o) จะเท่ากับจุดขยายเดิม

ถ้าจุดใหม่ที่แทรก ไม่เป็นจุดขยายเดิมคือเป็นจุดบนเส้น จุดเดิม (pos_o) จะเท่ากับ 0

แถวตารางเส้นถนนที่รถโดยสารผ่าน (row_m) คือ แถวของแถวตารางเส้นถนนที่รถโดยสารผ่าน (mapline) ที่จุดใหม่ไปแทรก

ชื่อโครงสร้าง ตารางคำสั่งชั่วคราวแทรกจุดในการขยาย (comroadm)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

การใช้งาน ใช้เก็บคำสั่งชั่วคราวแทรกจุดที่แปลงไป เพื่อนำไปปรับปรุงแฟ้มข้อมูลส่วนโค้งของถนน (MAPR.DAT)

ลักษณะโครงสร้าง

```
typedef struct {
        char com_t;     ประเภทคำสั่ง
        int pos_f;      จุดต้นช่วงถนน
        int pos_t;      จุดปลายช่วงถนน
        int link_m;     ตัวชี้ส่วนโค้ง
} COMROADM
```

รายละเอียด

ประเภทคำสั่ง (com_t) ได้แก่

การเพิ่มเติมเส้นทาง	ใช้ 'i'
การลบเส้นทาง	ใช้ 'd'

ตัวชี้ส่วนโค้ง (link_m) จะชี้แถวของชุดสายรายการจุดขยายที่ปรับใหม่ (acrcomroad)

ชื่อโครงสร้าง ชุดสายรายการจุดขยายที่ปรับใหม่ (acrcomroad)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

การใช้งาน ใช้บอกทางเดินของเส้นถนนที่รถโดยสารผ่านที่ถูกปรับใหม่เนื่องจากมีการแทรกจุด

ลักษณะโครงสร้าง

```
typedef struct mapacrnode {
        int  pos;
        struct mapacrnode *next;
} MAPACRNODE
```

ชื่อโครงสร้าง ตารางคำสั่งชั่วคราวแทรกจุดในเส้นทาง (comroadb)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

การใช้งาน ใช้เก็บคำสั่งชั่วคราวแทรกจุดที่แปลงไป เพื่อนำไปปรับปรุงกับแฟ้มข้อมูลเส้นทางเดินรถโดยสารประจำทาง (BUS.DAT)

ลักษณะโครงสร้าง

```
typedef struct {
        char com_t;      ประเภทคำสั่ง
        int pos_f;       จุดต้นช่วงถนน
        int pos_t;       จุดปลายช่วงถนน
        char type_m;     ประเภทถนน
        int row_m;       แถวตารางเส้นถนนที่รถโดยสารผ่าน
} COMROADB
```

รายละเอียด

ประเภทคำสั่ง (com_t) ได้แก่

การเพิ่มเติมเส้นทาง ใช้ 'i'

การลบเส้นทาง ใช้ 'd'

• แถวตารางเส้นถนนที่รถโดยสารผ่าน (row_m) เป็นแถวตารางเส้นถนนที่รถโดยสารผ่านที่ถูกแทรก จะใช้ในการอ่านสายรถที่ผ่านช่วงดังกล่าวทั้งหมด เมื่ออ่านข้อมูลสายรถแล้ว จะใช้เก็บแถวของชุดสายรายการสายรถที่ปรับใหม่ (buscomroad)

ชื่อโครงสร้าง ชุดสายรายการสายรถที่ปรับใหม่ (buscomroad)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

การใช้งาน ใช้เก็บสายรถที่ปรับใหม่ เนื่องจากมีการแทรกจุด

ลักษณะโครงสร้าง

```
typedef struct busnode {
        int  bus_n;
        struct busnode  *next;
} BUSNODE
```

ชื่อโครงสร้าง ตารางรหัสสายรถโดยสารประจำทาง (buskey)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

อ่านข้อมูลจาก แฟ้มข้อมูลสายรถโดยสารประจำทางแต่ละสาย (BUSKEY.DAT)

การใช้งาน 1. ใช้บอกรหัสสายรถ, สถานีต้นทาง และสถานีปลายทาง

2. ใช้เพิ่มเติมสายรถ

ลักษณะโครงสร้าง

```
typedef struct {
        int bus_n;      หมายเลขสายรถ
        int bus_k;      รหัสสายรถ
        int st_f;       สถานีต้นทาง
        int st_t;       สถานีปลายทาง
} BUSKEY
```

รายละเอียด

รหัสสายรถ (bus_k) เป็นคีย์หลัก (Primary key)

ชื่อโครงสร้าง ตารางรายละเอียดรถโดยสารประจำทาง (busname)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลเส้นทางเดินรถโดยสารประจำทางแต่ละสาย

อ่านข้อมูลจาก แฟ้มข้อมูลสายรถโดยสารประจำทางแต่ละสาย (BUSKEY.DAT)

การใช้งาน ใช้เพื่อเปรียบเทียบรายละเอียดสายรถ

ลักษณะโครงสร้าง

```
typedef struct {
        int bus_n;      หมายเลขสายรถ
        int bus_k;      รหัสสายรถ
        int st_f;       สถานีต้นทาง
        int st_t;       สถานีปลายทาง
        char bus_name[16]; รายละเอียดสายรถ
} BUSNAME
```

รายละเอียด

รหัสสายรถ (bus_k) เป็นคีย์หลัก (Primary key)

ชื่อโครงสร้าง ตารางจุดข้างเคียง (mapcon)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

การใช้งาน ใช้เก็บจุดข้างเคียง เมื่อทำการเคลื่อนย้ายจุดเส้นที่ต่อกับจุดข้างเคียงจะต้องเลื่อนตาม

ลักษณะโครงสร้าง

```
typedef struct {
        int pos;        จุดข้างเคียง
        char tpos;      ประเภทจุดข้างเคียง
        char type_m;    ประเภทเส้นถนน
} MAPCON
```

รายละเอียด

ถ้าจุดข้างเคียงเป็นจุดที่เกี่ยวกับเส้นทางเดินรถ ประเภทจุดข้างเคียงจะเป็น 'R'

ถ้าจุดข้างเคียงเป็นจุดขยาย ประเภทจุดข้างเคียงจะเป็น 'A'

ถ้าช่วงถนนระหว่างจุดข้างเคียงข้างกับจุดที่จะเคลื่อนย้ายเป็นถนนธรรมดา ประเภทเส้นถนนจะเป็น 'r'

ถ้าช่วงถนนระหว่างจุดข้างเคียงข้างกับจุดที่จะเคลื่อนย้ายเป็นถนนทางด่วน ประเภทเส้นถนนจะเป็น 'h'

ชื่อโครงสร้าง ตารางจุดเส้นประกอบข้างเคียง (mapcompcon)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลตำแหน่งของถนนและแม่น้ำ

การใช้งาน ใช้เก็บจุดเส้นประกอบข้างเคียง เมื่อทำการเคลื่อนย้ายจุด เส้นที่ต่อกับจุดเส้นประกอบข้างเคียงจะต้องเลื่อนตาม

ลักษณะโครงสร้าง

```
typedef struct {
        int pos_x;      ตำแหน่งแนวนอน
        int pos_y;      ตำแหน่งแนวตั้ง
        char type_m;    ประเภทของเส้น
} MAPCOMPCON
```

รายละเอียด

ถ้าเส้นที่ต่อระหว่างจุดเส้นประกอบข้างเคียงกับจุดที่เคลื่อนย้ายเป็น ถนนที่ไม่มีสายรถประจำทางผ่าน ประเภทของเส้นจะเป็น 'R'

ถ้าเส้นที่ต่อระหว่างจุดเส้นประกอบข้างเคียงกับจุดที่เคลื่อนย้ายเป็น เส้นประกอบแม่น้ำ ประเภทของเส้นจะเป็น 'V'

<u>ชื่อโครงสร้าง</u> ตารางรายชื่อสถานที่และสัญญลักษณ์ (stnameicn)

โปรแกรมที่ใช้งาน โปรแกรมปรับปรุงข้อมูลสถานที่และสร้างพื้นที่เพิ่ม

อ่านข้อมูลจาก 1. แฟ้มข้อมูลสถานที่สำคัญ (IMPORT.DAT)

2. แฟ้มข้อมูลโรงแรม (HOTEL.DAT)

3. แฟ้มข้อมูลโรงพยาบาล (HOSPITAL.DAT)

4. แฟ้มข้อมูลสถานีตำรวจ (POLICE.DAT)

5. แฟ้มข้อมูลห้างสรรพสินค้า (STORE.DAT)

การใช้งาน ใช้ปรับปรุงตำแหน่งและชื่อแฟ้มข้อมูลที่ใช้เก็บภาพสัญญลักษณ์

ลักษณะโครงสร้าง

```
typedef struct {
        char name_s[26];   ชื่อสถานที่
        int pos_x;         ตำแหน่งแนวนอน
        int pos_y;         ตำแหน่งแนวตั้ง
        char fname_s[13];  ชื่อแฟ้มข้อมูลที่ใช้เก็บภาพสัญญลักษณ์
} STNAMEICN
```

รายละเอียด

ชื่อสถานที่ (name_s) มีความยาวไม่เกิน 25 ตัวอักษร

ตำแหน่งแนวนอน (แกน X) คือ

pos_x = ตำแหน่งแนวนอน (แกน X) หยาบ * 10 +
ตำแหน่งแนวนอน (แกน X) ละเอียด

ตำแหน่งแนวตั้ง (แกน Y) คือ

pos_y = ตำแหน่งแนวตั้ง (แกน Y) หยาบ * 10 +
ตำแหน่งแนวตั้ง (แกน Y) ละเอียด

ถ้าแต่ข้อมูลชื่อสถานที่สำคัญ แต่ยังไม่มีตำแหน่งในวางตำแหน่งในแผนที่ตำแหน่งแนวนอน (pos_x) เป็น -1

ชื่อแฟ้มข้อมูลที่ใช้เก็บภาพสัญญลักษณ์ (fname_s) ชื่อของแฟ้มข้อมูลจะต้องมีนามสกุลเป็น ICN และมีความยาวไม่เกิน 12 ตัวอักษร

ถ้าแต่ข้อมูลชื่อสถานที่ แต่ยังไม่มีตำแหน่งในวางตำแหน่งในแผนที่ตำแหน่งแนวนอน (pos_x) เป็น -1

ประวัติผู้เขียน

ผู้เขียนชื่อ นาย วศิน สินธุภิญโญ เกิดเมื่อวันที่ 5 มีนาคม พ.ศ. 2512 ที่จังหวัดนครราชสีมา สำเร็จการศึกษาปริญญาวิทยาศาสตร์บัณฑิต (คณิตศาสตร์) เกียรตินิยมอันดับ สอง จาก มหาวิทยาลัยขอนแก่น