

ภาคผนวก ก.  
รหัสเทียมของโปรแกรม CU-SWMM

รหัสเทียบของ CUSWMM.BAT

## โปรแกรมหลัก

ปิดการแสดงชื่อโปรแกรม

วนรอบ

เรียกโปรแกรม CUMENU.EXE

ถ้าหาโปรแกรม PROCESS.BAT ไม่พบ

ออกจากโปรแกรม

เรียกโปรแกรม PROCESS.BAT

จบ

**รหัสเทียมของ CUMENU.C**



ชื่อโปรแกรม 1-4 = EDIT, SWMM, COST, EDIT

### โปรแกรมหลัก

ลบจอภาพ

สร้าง pop-up menu

จำแนก ตัวเลือกจาก pop-up menu

กรณี = 1-4 : สร้าง process (ตัวเลือกจาก pop-up menu, ชื่อโปรแกรม 1-4)

กรณี = 5 : ลบไฟล์ PROCESS.BAT

ลบ pop-up menu

ลบจอภาพ

ออกจากโปรแกรม

### สร้าง process (ตัวเลือก, ชื่อโปรแกรม)

เปิดไฟล์ PROCESS.BAT เพื่อเขียน

ถามชื่อไฟล์

สร้างนามสกุลไฟล์ข้อมูลเข้า, ข้อมูลออก

จำแนก ตัวเลือก

กรณี = 1 : พิมพ์ “ชื่อโปรแกรม ชื่อไฟล์ข้อมูลเข้า” ลงไฟล์ PROCESS.BAT

กรณี = 2 : พิมพ์ “ @ echo off \n “

“if not exist ชื่อไฟล์ข้อมูลเข้า goto process-fail \n”

“ชื่อโปรแกรม ชื่อไฟล์ข้อมูลเข้า ชื่อไฟล์ข้อมูลออก \n”

“ : process \_ fail \n”

“ @ echo Data file doesn’t exist. \n”

“ : end-process \n”

“ pause” ลงไฟล์ PROCESS.BAT

กรณี = 3 : พิมพ์ “ @ echo off \n “

“If not exist ไฟล์ข้อมูลเข้า goto process\_fail \n”

“ชื่อโปรแกรม ไฟล์ข้อมูลเข้า \n”

“goto end\_process \n”

“@ echo Data file doesn’t exist.\n”

“pause \n”

“ : end\_process” ลงไฟล์ PROCESS.BAT

กรณี = 4 : พิมพ์ “ @ echo off \n “

```

“If not exist ชื่อไฟล์ข้อมูลออก goto process_fail \n”
“ชื่อโปรแกรม ไฟล์ข้อมูลออก \n”
“goto end_process \n”
“ : process_fail \n”
“ @ echo Output file doesn’t exist. \n”
“pause \n”
“ : end_process” ลงไฟล์ PROCESS.BAT

```

ปิดไฟล์ PROCESS.BAT

จบ

### สร้าง pop-up menu (จำนวนตัวเลือก, อนุกรมของตัวเลือก)

สร้างกรอบ menu

พิมพ์ตัวเลือกตัวแรกด้วย REVERSE

พิมพ์ตัวเลือกที่เหลือด้วย NORMAL

ตัวเลือก = 1

วนรอบ

รอการกดแป้นพิมพ์

จำแนก แป้นพิมพ์ที่กด

กรณี ENTER = กลับและส่งค่าตัวเลือกกลับ

กรณี ลูกศรขึ้น = เลื่อนตัวเลือกขึ้น (จำนวนตัวเลือก, ตัวเลือก)

กรณี ลูกศรลง = เลื่อนตัวเลือกลง (จำนวนตัวเลือก, ตัวเลือก)

จบ

### เลื่อนตัวเลือกขึ้น (จำนวนตัวเลือก, ตัวเลือก)

พิมพ์ตัวเลือกด้วย NORMAL

ถ้า ตัวเลือก = จำนวนตัวเลือก

ตัวเลือก = 1

ถ้าไม่

ลดค่าตัวเลือกลง 1

พิมพ์ตัวเลือกด้วย REVERSE

กลับและส่งค่าตัวเลือกกลับ

## เลื่อนตัวเลือกลง (จำนวนตัวเลือก, ตัวเลือก)

พิมพ์ตัวเลือกด้วย NORMAL

ถ้า  $\text{ตัวเลือก} = \text{จำนวนตัวเลือก}$

$\text{ตัวเลือก} = 1$

ถ้าไม่

ลดค่าตัวเลือกลง 1

พิมพ์ตัวเลือกด้วย REVERSE

กลับและลดค่าตัวเลือกกลับ

รหัสเทียมของ COST.C

## โปรแกรมหลัก (ไฟล์ข้อมูลเข้า)

ลบจอภาพ

พิมพ์หน้าจอตามต้องการ

เปิดไฟล์ข้อมูลเข้า

ถ้า ตรวจหา Transport block (ไฟล์ข้อมูลเข้า) ไม่สำเร็จ

ปิดไฟล์ข้อมูลเข้า

ออกจากโปรแกรม

ปิดไฟล์ข้อมูลเข้า

ตั้งค่าราคา

วนรอบ

รอการกดแป้นพิมพ์

จำแนก แป้นพิมพ์ที่กด

กรณี = F1 : อ่านข้อมูลราคาจาก disk

กรณี = F2 : เขียนข้อมูลราคาลง disk

กรณี = F3 : พิมพ์ราคารวม

กรณี = F4 : ออกจากโปรแกรม

กรณี = F5 : สร้างข้อมูลราคาต่อกลม

กรณี = F6 : สร้างข้อมูลราคาต่อสี่เหลี่ยม

กรณี = F7 : สร้างข้อมูลราคางานโยธา

กรณี = F9 : ประเมินราคาแต่ละช่วงทางระบายน้ำ

กรณี = F10 : ประเมินราคารวม

จบ

## ตรวจหา Transport Block (ไฟล์ข้อมูลเข้า)

วนรอบ

ถ้า อ่านข้อความจากไฟล์ข้อมูลเข้า 1 บรรทัด ไม่ได้

พิมพ์ว่าไม่พบ Transport block

กลับและส่งค่า 0 กลับ

หากยังหาข้อความ “\$TRANSPORT” ในข้อความที่อ่านไม่พบ

พิมพ์ว่าพบ Transport block

วนรอบ

ถ้า อ่านข้อความจากไฟล์ข้อมูลเข้า 1 บรรทัด ไม่ได้

พิมพ์ว่าไม่พบทางระบายน้ำ

กลับและส่งค่า 0 กลับ  
 หากยังหาข้อความ “E1” ในข้อความที่อ่านไม่พบ  
 วนรอบ

ถ้า อ่านค่าหมายเลข, ชนิด, ความยาว และขนาดของทางระบายน้ำไม่ได้  
 พิมพ์ว่ามีทางระบายน้ำบางส่วน ผิดพลาด  
 กลับและส่งค่า 0 กลับ

หากยังอ่านข้อความจากไฟล์ข้อมูลเข้า 1 บรรทัดไม่ได้ และ หาข้อความ “E1” ในข้อความที่  
 อ่านไม่พบ

พิมพ์ว่า พบทางระบายน้ำ ..... ช่วง  
 กลับและส่งค่า 1 กลับ

### ตั้งค่าราคา

ตั้งค่าขนาดเส้นผ่านศูนย์กลางของท่อกลมให้ครบทุกขนาด  
 ตั้งค่าขนาดความกว้าง, สูงของท่อสี่เหลี่ยมให้ครบทุกขนาด

วนรอบ ตามจำนวนขนาดของท่อกลม

ราคาท่อกลม = 0

วนรอบ ตามจำนวนขนาดของท่อสี่เหลี่ยม

ราคาท่อสี่เหลี่ยม = 0

วนรอบ ตามจำนวนช่วงของทางระบายน้ำ

ราคาทางระบายน้ำ = 0

พิมพ์ว่า ตั้งราคาเสร็จเรียบร้อยแล้ว

จบ

### สร้างข้อมูลราคาท่อกลม

ถ้า ค้นหาท่อกลมในไฟล์ข้อมูลเข้าไม่พบ

กลับ

เก็บจอภาพ

ลบจอภาพ

พิมพ์หน้าจอตามต้องการ

วนรอบ ตามจำนวนขนาดของท่อกลม

พิมพ์ ขนาดของท่อกลม

วนรอบ ตามจำนวนขนาดของท่อกลม

อ่าน ราคาท่อกลม

พิมพ์ ราคาต่อกลม  
 คั้นจอภาพ  
 จบ

### สร้างข้อมูลราคาท่อสี่เหลี่ยม

ถ้า ค้นหาท่อสี่เหลี่ยมในไฟล์ข้อมูลเข้าไม่พบ  
 กลับ  
 เก็บจอภาพ  
 ลบจอภาพ  
 พิมพ์หน้าจอตามต้องการ  
 วนรอบ ตามจำนวนขนาดของท่อสี่เหลี่ยม  
     พิมพ์ขนาดของท่อสี่เหลี่ยม  
 วนรอบ ตามจำนวนขนาดของท่อสี่เหลี่ยม  
     อ่าน ราคาท่อสี่เหลี่ยม  
     พิมพ์ ราคาท่อสี่เหลี่ยม  
 คั้นจอภาพ  
 จบ

### สร้างข้อมูลราคางานโยธา

เก็บจอภาพ  
 ลบจอภาพ  
 พิมพ์ ราคางานขุด  
 พิมพ์ ราคางานถม  
 พิมพ์ ราคาเข็มพืด  
 พิมพ์ ราคาดาดคอนกรีต  
 พิมพ์ ราคาซ่อมผิวจราจร  
 พิมพ์ ความลึกดินถมต่ำสุด  
 พิมพ์ ช่องว่างเหนือผิวน้ำต่ำสุด  
 พิมพ์ ความชันด้านของการขุด  
 พิมพ์ ความลึกที่ต้องใช้เข็มพืด  
 อ่าน ราคางานขุด  
 พิมพ์ ราคางานขุด  
 อ่าน ราคางานถม

พิมพ์	ราคางานถม
อ่าน	ราคาเข็มพืด
พิมพ์	ราคาเข็มพืด
อ่าน	ราคาตาดคอนกรีต
พิมพ์	ราคาตาดคอนกรีต
อ่าน	ราคาซ่อมผิวจราจร
พิมพ์	ราคาซ่อมผิวจราจร
อ่าน	ความลึกดินถมต่ำสุด
พิมพ์	ความลึกดินถมต่ำสุด
อ่าน	ช่องว่างเหนือผิวน้ำต่ำสุด
พิมพ์	ช่องว่างเหนือผิวน้ำต่ำสุด
อ่าน	ความชันด้านข้างของการขุด
พิมพ์	ความชันด้านข้างของการขุด
อ่าน	ความลึกที่ต้องใช้เข็มพืด
พิมพ์	ความลึกที่ต้องใช้เข็มพืด
จบ	

### ประเมินราคา

เก็บจอบภาพ

ลบจอบภาพ

พิมพ์ “ราคารวมทางระบายน้ำ”

พิมพ์ “ราคารวมงานถม”

พิมพ์ “ราคารวมเข็มพืด”

พิมพ์ รวมราคาตาดคอนกรีต

พิมพ์ “ราคารวมซ่อมผิวจราจร”

กำหนดราคารวมทุกประเภท = 0

วนรอบ ตามจำนวนทางระบายน้ำ

จำแนกชนิดของทางระบายน้ำ

กรณีท่อกลม = คำนวณราคาท่อกลม

กรณีท่อเหลี่ยม = คำนวณราคาท่อสี่เหลี่ยม

กรณีรางคางหมู = คำนวณราคารางคางหมู

พิมพ์ราคารวมทุกประเภท

ราคารวม = ราคารวมทุกประเภทรวมกัน

พิมพ์ราคารวม

คืนจอบภาพ



จบ

คำนวณราคาท่อกลม

วนรอบตามจำนวนขนาดของท่อกลม

ถ้าขนาดทางระบายน้ำ = ขนาดท่อกลม

$$\text{ราคารวมทางระบายน้ำที่เพิ่ม} = \text{ความยาวท่อกลม} \times \text{ราคาต่อหน่วยความยาวของท่อกลม}$$

$$\text{ความลึก} = \text{ขนาดเส้นผ่านศูนย์กลางท่อกลม} + \text{ความลึกดินถมต่ำสุด}$$

$$\text{ความกว้างที่ห้อง} = \text{ขนาดเส้นผ่านศูนย์กลางท่อกลม} + 1$$

$$\text{ถ้าความลึก} \geq \text{ความลึกที่ต้องใช้เข็มพืด}$$

$$\text{ความกว้างที่ปาก} = \text{ความกว้างที่ห้อง}$$

$$\text{ราคาเข็มพืดที่เพิ่ม} = \text{ความยาวของท่อกลม} \times \text{ราคาต่อหน่วยความยาวเข็มพืด}$$

ถ้าไม่

$$\text{ความกว้างที่ปาก} = \text{ความกว้างที่ห้อง} + 2 (\text{ความลึก} \times \text{ความชันด้านข้างของการขุด})$$

$$\text{หน้าตัดการขุด} = 0.5 \times \text{ความลึก} \times (\text{ความกว้างที่ปาก} + \text{ความกว้างที่ห้อง})$$

$$\text{หน้าตัดท่อกลม} = \pi \times \text{ขนาดเส้นผ่านศูนย์กลางท่อกลม}^2 / 4$$

$$\text{ราคารวมงานขุดที่เพิ่ม} = \text{หน้าตัดการขุด} \times \text{ราคาต่อหน่วยปริมาตรของงานขุด}$$

$$\text{ราคารวมงานถมที่เพิ่ม} = (\text{หน้าตัดการขุด} - \text{หน้าตัดท่อกลม}) \times \text{ราคาต่อหน่วยปริมาตรของการถม}$$

$$\text{ราคาซ่อมผิวจราจรที่เพิ่ม} = \text{ความกว้างที่ปาก} \times \text{ความยาวท่อกลม} \times \text{ราคาต่อหน่วยพื้นที่ของการซ่อมผิวจราจร}$$

จบ

คำนวณราคาท่อสี่เหลี่ยม

วนรอบตามจำนวนขนาดของท่อสี่เหลี่ยม

ถ้าขนาดทางระบายน้ำ = ขนาดท่อสี่เหลี่ยม

$$\text{ราคารวมทางระบายน้ำที่เพิ่ม} = \text{ความยาวท่อสี่เหลี่ยม} \times \text{ราคาต่อหน่วยความยาวของท่อสี่เหลี่ยม}$$

$$\text{ความลึก} = \text{ความสูงท่อสี่เหลี่ยม} + \text{ความลึกดินถมต่ำสุด}$$

$$\text{ความกว้างที่ท้อง} = \text{ความกว้างท่อสี่เหลี่ยม} + 1$$

ถ้าความลึก  $\geq$  ความลึกที่ต้องใช้เชื่อมพืด

$$\text{ความกว้างที่ปาก} = \text{ความกว้างที่ท้อง}$$

$$\text{ราคาเชื่อมพืดที่เพิ่ม} = \text{ความยาวท่อสี่เหลี่ยม} \times \text{ราคาต่อหน่วยความยาวของเชื่อมพืด}$$

ถ้าไม่

$$\text{ความกว้างที่ปาก} = \text{ความกว้างที่ท้อง} + 2 (\text{ความลึก} \times \text{ความชันด้านข้างของการขุด})$$

$$\text{หน้าตัดการขุด} = 0.5 \times \text{ความลึก} \times (\text{ความกว้างที่ปาก} + \text{ความกว้างที่ท้อง})$$

$$\text{หน้าตัดท่อสี่เหลี่ยม} = \text{ความกว้างท่อสี่เหลี่ยม} \times \text{ความสูงท่อสี่เหลี่ยม}$$

$$\text{รวมราคางานขุดที่เพิ่ม} = \text{หน้าตัดการขุด} \times \text{ราคาต่อหน่วยปริมาตรของงานขุด}$$

$$\text{ราคารวมงานถมที่เพิ่ม} = (\text{หน้าตัดการขุด} - \text{หน้าตัดท่อกลม}) \times \text{ราคาต่อหน่วย}$$

ปริมาตรของการถม

$$\text{ราคาซ่อมผิวจราจรที่เพิ่ม} = \text{ความกว้างที่ปาก} \times \text{ความยาวท่อสี่เหลี่ยม} \times \text{ราคาต่อหน่วยพื้นที่ของการซ่อมผิวจราจร}$$

จบ

### คำนวณราคารางคางหมู

$$\text{ความลึก} = \text{ความสูงน้ำ} + \text{ช่องว่างเหนือผิวน้ำต่ำสุด}$$

$$\text{ความกว้างที่ปาก} = \text{ความกว้างท้องราง} + 2 (\text{ความลึก} \times \text{ความชันด้านข้างราง})$$

$$\text{หน้าตัดการขุด} = 0.5 \times \text{ความลึก} \times (\text{ความกว้างที่ปาก} + \text{ความกว้างท้องราง})$$

$$\text{ราคารวมงานขุดที่เพิ่ม} = \text{หน้าตัดการขุด} \times \text{ราคาต่อหน่วยปริมาตรของการขุด}$$

$$\text{ราคารวมคอนกรีตที่เพิ่ม} = \text{เส้นรอบรูปรางคางหมู} \times \text{ความยาวรางคางหมู} \times \text{ราคาต่อหน่วยการตาดคอนกรีต}$$

### พิมพ์ราคารวมทุกประเภท (ราคารวมทุกประเภท)

พิมพ์ราคารวมทางระบายน้ำ

พิมพ์ราคารวมงานขุด

พิมพ์ราคารวมงานถม

พิมพ์ราคารวมเชื่อมพืด

พิมพ์ราคากรมตาดคอนกรีต  
พิมพ์ราคากรมซ่อมผิวจราจร

ภาคผนวก ข.  
รหัสต้นฉบับของโปรแกรม CU-SWMM

**รหัสต้นฉบับของ CUSWMM.BAT**

```
@echo off
:start_run
cmenu
if not exist process.bat goto end_run
call process.bat
goto start_run
:end_run
```

รหัสต้นฉบับของ CUMENU.C

```

#include <bios.h>
#include <conio.h>
#include <dos.h>
#include <errno.h>
#include <io.h>
#include <alloc.h>
#include <process.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define MAXWND      16
#define LEFTTOP     218
#define LEFTBOTTOM  192
#define RIGHTTOP    191
#define RIGHTBOTTOM 217
#define HORIZON     196
#define VERTICAL    179
#define CR          0x001C
#define UP          0x0048
#define DN          0x0050
#define CHOICENO    5
#define MINX        1
#define MINY        1
#define MAXX        80
#define MAXY        25
#define NORMAL      7
#define REVERSE     112

int Wnum = 0;
int Wactive = 0;
char far *Vdomem = (char far *) 0xB8000000;
char *Tagcuswmm = " CU-SWMM Version 1.0
char *Hlpcuswmm = " Use Up and Down arrow keys to select process ...

char *Choice[CHOICENO] = {
    " Data Editor  ",
    " Simulation  ",
    " Cost Estimate ",
    " View Output  ",
    " Exit        ",
};

char *Process[CHOICENO-1] = {
    "edit",
    "swmm",
    "cost",
    "edit"
};

struct Wndow {
    int *buffer;
    int left,top,right,bottom;
    int curx,cury;
    int frmattr,wndattr;
    char *title;
    int titattr;
} Wnd[MAXWND];

void process(int choice,char *s);
void cursorsize(int startline,int stopline);
void clrscr(void);
void writech(int x,int y,unsigned char ch,int attr);
void writes(int x,int y,unsigned char *s,int attr);
int up(int chc,int left,int top,int wndattr,int chcattr,int choiceno,
char *choice[]);
int down(int chc,int left,int top,int wndattr,int chcattr,
int choiceno,char *choice[]);
void winit(int num,int left,int top,int right,int bottom,char *title,
int frmattr,int wndattr,int titattr);
void wopen(int num);
void wcursor(int x,int y);
int popup(int num,int chcattr,int choiceno,char *choice[]);
void wclose(void);

void main()
{
    int choice;

    winit(1,32,8,48,14," Menu ",NORMAL,NORMAL,REVERSE);
    clrscr();
    cursorsize(-1,-1);
    writes(MINX,MINY,Tagcuswmm,REVERSE);

```

Chulalongkorn University ";  
";



```

writes(MINX,MAXY,Hlpcuswmm,REVERSE);
choice = popup(1,REVERSE,CHOICENO,Choice);
switch (choice) {
    case 1 :
    case 2 :
    case 3 :
    case 4 : cursorsize(13,14);
              process(choice,Process[choice-1]);
              break;
    case 5 : cursorsize(13,14);
              remove("process.bat");
}
wclose();
clrscr();
gotoxy(1,1);
}

void process(int choice,char *s)
{
    FILE *fp;
    char str[9];
    char fnamedat[13],fnameout[13];
    char *dat = ".dat";
    char *out = ".out";

    fp = fopen("process.bat","w");
    writes(30,16," File name :",NORMAL);
    gotoxy(43,16);
    gets(str);
    strcpy(fnamedat,str);
    strcpy(fnameout,str);
    strcat(fnamedat,dat);
    strcat(fnameout,out);
    switch (choice) {
        case 1 : fprintf(fp,"%s %s",s,fnamedat);
                  break;
        case 2 : fprintf(fp,"@echo off\n");
                  fprintf(fp,"if not exist %s goto process_fail\n",fnamedat);
                  fprintf(fp,"%s %s %s\n",s,fnamedat,fnameout);
                  fprintf(fp,"goto end_process\n");
                  fprintf(fp,":process_fail\n@echo Data file doesn't exist.\n");
                  fprintf(fp,":end_process\npause");
                  break;
        case 3 : fprintf(fp,"@echo off\n");
                  fprintf(fp,"if not exist %s goto process_fail\n",fnamedat);
                  fprintf(fp,"%s %s\n",s,fnamedat);
                  fprintf(fp,"goto end_process\n");
                  fprintf(fp,":process_fail\n@echo Data file doesn't exist.\n");
                  fprintf(fp,"pause\n");
                  fprintf(fp,":end_process");
                  break;
        case 4 : fprintf(fp,"@echo off\n");
                  fprintf(fp,"if not exist %s goto process_fail\n",fnameout);
                  fprintf(fp,"%s %s\n",s,fnameout);
                  fprintf(fp,"goto end_process\n");
                  fprintf(fp,":process_fail\n@echo Output file doesn't exist.\n");
                  fprintf(fp,"pause\n");
                  fprintf(fp,":end_process");
    }
    fclose(fp);
    return;
}

void cursorsize(int startline,int stopline)
{
    union REGS r;

    r.h.ah = 1;
    r.h.ch = startline;
    r.h.cl = stopline;
    int86(0x10,&r,&r);
}

void clrscr(void)
{
    union REGS r;

    r.h.ah = 6;
    r.h.al = 0;
    r.h.ch = 0;
    r.h.cl = 0;
    r.h.dh = 24;
    r.h.dl = 79;
}

```



```
r.h.bh = 7;
int86(0x10,&r,&r);
}

void writech(int x,int y,unsigned char ch,int attr)
{
    char far *m;

    m = Vdomem + ((y-1)*160) + (x-1)*2;
    *m++ = ch;
    *m = attr;
}

void writes(int x,int y,unsigned char *s,int attr)
{
    register char i;
    char far *m;

    m = Vdomem + ((y-1)*160) + (x-1)*2;
    for (i = x; *s ;i++) {
        *m++ = *s++;
        *m++ = attr;
    }
}

int up(int chc,int left,int top,int wndattr,int chcatr,int choiceno,
    char *choice[])
{
    writes(left+1,top+chc,choice[chc-1],wndattr);
    if (chc == 1)
        chc = choiceno;
    else
        chc--;
    writes(left+1,top+chc,choice[chc-1],chcatr);
    return(chc);
}

int down(int chc,int left,int top,int wndattr,int chcatr,int choiceno,
    char *choice[])
{
    writes(left+1,top+chc,choice[chc-1],wndattr);
    if (chc == choiceno)
        chc = 1;
    else
        chc++;
    writes(left+1,top+chc,choice[chc-1],chcatr);
    return(chc);
}

void winit(int num,int left,int top,int right,int bottom,char *title,
    int frmattr,int wndattr,int titattr)
{
    Wnd[num-1].buffer = malloc((right-left+1) * (bottom-top+1) * 2);
    Wnd[num-1].left = left;
    Wnd[num-1].top = top;
    Wnd[num-1].right = right;
    Wnd[num-1].bottom = bottom;
    Wnd[num-1].curx = 1;
    Wnd[num-1].cury = 1;
    Wnd[num-1].frmattr = frmattr;
    Wnd[num-1].wndattr = wndattr;
    Wnd[num-1].title = malloc(right - left);
    Wnd[num-1].title = title;
    Wnd[num-1].titattr = titattr;
    Wnum++;
}

void wopen(int num)
{
    register char i;
    int x;

    gettext(Wnd[num-1].left,Wnd[num-1].top,Wnd[num-1].right,Wnd[num-1].bottom,
        Wnd[num-1].buffer);
    writech(Wnd[num-1].left,Wnd[num-1].top,LEFTTOP,Wnd[num-1].frmattr);
    writech(Wnd[num-1].right,Wnd[num-1].top,RIGHTTOP,Wnd[num-1].frmattr);
    writech(Wnd[num-1].left,Wnd[num-1].bottom,LEFTBOTTOM,Wnd[num-1].frmattr);
    writech(Wnd[num-1].right,Wnd[num-1].bottom,RIGHTBOTTOM,Wnd[num-1].frmattr);
    for (i = Wnd[num-1].left + 1 ; i < Wnd[num-1].right ; i++) {
        writech(i,Wnd[num-1].top,HORIZON,Wnd[num-1].frmattr);
        writech(i,Wnd[num-1].bottom,HORIZON,Wnd[num-1].frmattr);
    }
    for (i = Wnd[num-1].top + 1 ; i < Wnd[num-1].bottom ; i++) {
```

```

    writech(Wnd[num-1].left,i,VERTICAL,Wnd[num-1].frmattr);
    writech(Wnd[num-1].right,i,VERTICAL,Wnd[num-1].frmattr);
}
x = Wnd[num-1].right - Wnd[num-1].left + 1 - strlen(Wnd[num-1].title);
x = Wnd[num-1].left + x / 2;
writes(x,Wnd[num-1].top,Wnd[num-1].title,Wnd[num-1].titattr);
wcursor(Wnd[num-1].curx,Wnd[num-1].cury);
Wactive = num;
}

void wcursor(int x,int y)
{
    Wnd[Wactive-1].curx = x;
    Wnd[Wactive-1].cury = y;
    gotoxy(Wnd[Wactive-1].left+x,Wnd[Wactive-1].top+y);
}

int popup(int num,int chcatr,int choiceno,char *choice[])
{
    register int chc;
    union key {
        char ch[2];
        int i;
    } c;

    wopen(num);
    writes(Wnd[num-1].left+1,Wnd[num-1].top+1,choice[0],chcatr);
    for (chc = 2 ; chc <= choiceno ; chc++)
        writes(Wnd[num-1].left+1,Wnd[num-1].top+chc,choice[chc-1],
            Wnd[num-1].wndattr);
    chc = 1;
    for (;;) {
        while (!bioskey(1));
        c.i = bioskey(0);
        switch (c.ch[1]) {
            case CR : return(chc);
            case UP : chc = up(chc,Wnd[num-1].left,Wnd[num-1].top,
                Wnd[num-1].wndattr,chcatr,choiceno,choice);
                break;
            case DN : chc = down(chc,Wnd[num-1].left,Wnd[num-1].top,
                Wnd[num-1].wndattr,chcatr,choiceno,choice);
                break;
        }
    }
}

void wclose(void)
{
    puttext(Wnd[Wactive-1].left,Wnd[Wactive-1].top,Wnd[Wactive-1].right,
        Wnd[Wactive-1].bottom,Wnd[Wactive-1].buffer);
    free(Wnd[Wactive-1].buffer);
}

```

**รหัสต้นฉบับของ COST.C**

```

#include <conio.h>
#include <dos.h>
#include <math.h>
#include <stdio.h>
#include <string.h>

#define MINX      1
#define MINY      1
#define STARTX    1
#define STARTY    2
#define ENDX      80
#define ENDY      24
#define MAXX      80
#define MAXY      25
#define NORMAL    7
#define REVERSE   112
#define UP        0x0048
#define DN        0x0050
#define F1        0x003B
#define F2        0x003C
#define F3        0x003D
#define F4        0x003E
#define F5        0x003F
#define F6        0x0040
#define F7        0x0041
#define F9        0x0043
#define F10       0x0044
#define MAXELEMENT 200
#define MAXDIA    12
#define MAXDIM    13
#define PI        3.141590654

int   Elenum = 0;
int   Eleno[MAXELEMENT];
int   Eletype[MAXELEMENT];
double Elelen[MAXELEMENT];
double Elegeo1[MAXELEMENT];
double Elegeo2[MAXELEMENT];
double Elegeo3[MAXELEMENT];

double Elecost[MAXELEMENT];
double Exccost[MAXELEMENT];
double Bkfcost[MAXELEMENT];
double Shpcost[MAXELEMENT];
double Lngcost[MAXELEMENT];
double Prpcost[MAXELEMENT];
double Telecost = 0.0;
double Texccost = 0.0;
double Tbkfcost = 0.0;
double Tshpcost = 0.0;
double Tlngcost = 0.0;
double Tprpcost = 0.0;
double Totalcost = 0.0;
double Uexccost = 0.0;
double Ubkfcost = 0.0;
double Ushpcost = 0.0;
double Ulngcost = 0.0;
double Uprpcost = 0.0;
double Cover = 0.6;
double Freebrd = 0.3;
double Sslope = 1.0;
double Shpdepth = 3.0;
int   Elest = 0;
int   Totest = 0;

char *Tagcost = " Cost Estimate
char *Hlpcost1 = " F1:Load F2:Save F3:Print F4:Exit    F5:Circ F6:Rect F7:Civ    CU-SWMM Version 1.0 ";
char *Hlpcost2 = " Please enter data of each item ...      F9:Ele F10:Total ";
char *Hlpcost3 = " Press any key to return ... ";
char *Hlpcost4 = " Use UP and DOWN arrow keys to scroll ... ";
char far *Vdomem = (char far *) 0xB8000000;      F4:Exit F10:Total ";

struct Circcost {
    double dia;
    double cost;
} Circ[MAXDIA];

struct Rectcost {

```

```

    double w;
    double h;
    double cost;
} Rect[MAXDIM];

int scantrn(FILE *fp);
void init_trncost(void);
void loadcost(void);
void savecost(void);
void printcost(void);
void edit_circrcost(void);
void edit_rectcost(void);
void edit_civlcost(void);
void elementcost(void);
void totalcost(void);
void circrcost(int i);
void rectcost(int i);
void trapcost(int i);
void cursorsize(int startline,int stopline);
void writech(int x,int y,unsigned char ch,int attr);
void writes(int x,int y,unsigned char *s,int attr);
void clrline(int line,int attr);
void scrollup(int left,int top,int right,int bottom,int lineno,int attr);
void scrolldn(int left,int top,int right,int bottom,int lineno,int attr);

void main(int argc,char *argv[])
{
    FILE *fp;
    union key {
        char ch[2];
        unsigned i;
    } c;

    clrscr();
    writes(MINX,MINY,Tagcost,REVERSE);
    fp = fopen(argv[1],"r");
    cursorsize(-1,-1);
    if (!scantrn(fp)) {
        fclose(fp);
        getch();
        exit(0);
    }
    fclose(fp);
    init_trncost();
    writes(STARTX,STARTY+8,"Use function key to select process ...",NORMAL);
    for (;;) {
        writes(MINX,MAXY,Hlpcost1,REVERSE);
        while (!bioskey(1));
        c.i = bioskey(0);
        if (!c.ch[0]) {
            switch (c.ch[1]) {
                case F1 : loadcost();
                    break;
                case F2 : savecost();
                    break;
                case F3 : printcost();
                    break;
                case F4 : exit(1);
                case F5 : edit_circrcost();
                    break;
                case F6 : edit_rectcost();
                    break;
                case F7 : edit_civlcost();
                    break;
                case F9 : elementcost();
                    break;
                case F10 : totalcost();
                    break;
            }
        }
    }
}

int scantrn(FILE *fp)
{
    register int i;
    char str[83];

```

```

do {
    if (fgets(str,82,fp) == NULL) {
        writes(STARTX,STARTY+2,"No TRANSPORT block founded.",NORMAL);
        return(0);
    }
} while (strstr(str,"$TRANSPORT") == NULL);
writes(STARTX,STARTY+2,"TRANSPORT block founded.",NORMAL);
do {
    if (fgets(str,82,fp) == NULL) {
        writes(STARTX,STARTY+4,"No element founded.",NORMAL);
        return(0);
    }
} while (strstr(str,"E1") == NULL);
i = 1;
do {
    if (sscanf(str,"%s %d %d %d %d %d %lf %lf %*f %*f %lf %*f %lf",
        &Eleno[i-1],&Eletype[i-1],&Elelen[i-1],&Elegeo1[i-1],
        &Elegeo2[i-1],&Elegeo3[i-1]) != 6) {
        writes(STARTX,STARTY+4,"Some elements are not correct.",NORMAL);
        return(0);
    }
    i++;
} while (fgets(str,82,fp)!=NULL && strstr(str,"E1")!=NULL);
Elenum = i-1;
gotoxy(STARTX,STARTY+4);
cprintf("%d elements founded.",Elenum);
return(1);
}

void init_trncost(void)
{
    FILE *fp;
    register int i;

    Circ[0].dia = 0.30;   Rect[0].w = 1.20;   Rect[0].h = 1.20;
    Circ[1].dia = 0.40;   Rect[1].w = 1.50;   Rect[1].h = 1.20;
    Circ[2].dia = 0.50;   Rect[2].w = 1.50;   Rect[2].h = 1.50;
    Circ[3].dia = 0.60;   Rect[3].w = 1.80;   Rect[3].h = 1.50;
    Circ[4].dia = 0.80;   Rect[4].w = 1.80;   Rect[4].h = 1.80;
    Circ[5].dia = 1.00;   Rect[5].w = 2.10;   Rect[5].h = 1.80;
    Circ[6].dia = 1.20;   Rect[6].w = 2.10;   Rect[6].h = 2.10;
    Circ[7].dia = 1.50;   Rect[7].w = 2.40;   Rect[7].h = 2.10;
    Circ[8].dia = 1.75;   Rect[8].w = 2.40;   Rect[8].h = 2.40;
    Circ[9].dia = 2.00;   Rect[9].w = 2.70;   Rect[9].h = 2.40;
    Circ[10].dia = 2.25;  Rect[10].w = 2.70;  Rect[10].h = 2.70;
    Circ[11].dia = 2.50;  Rect[11].w = 3.00;  Rect[11].h = 2.70;
                                Rect[12].w = 3.00;  Rect[12].h = 3.00;

    for (i = 1;i <= MAXDIA;i++)
        Circ[i-1].cost = 0;
    for (i = 1;i <= MAXDIM;i++)
        Rect[i-1].cost = 0;
    for (i = 1;i <= Elenum;i++) {
        Eleccost[i-1] = 0;
        Exccost[i-1] = 0;
        Bkfcost[i-1] = 0;
        Shpcost[i-1] = 0;
        Lngcost[i-1] = 0;
        Prpcost[i-1] = 0;
    }
    writes(STARTX,STARTY+6,"Initialize cost complete.",NORMAL);
}

void loadcost(void)
{
    register int i;
    FILE *fp;
    char str[9];
    char fnamecst[13];
    char *cst = ".cst";

    cursorsize(13,14);
    writes(STARTX,ENDY-2,"Cost file to load :",NORMAL);
    gotoxy(STARTX+20,ENDY-2);
    scanf("%s",str);
    clrline(ENDY-2,NORMAL);
    strcpy(fnamecst,str);
    strcat(fnamecst,cst);
    if ((fp=fopen(fnamecst,"r")) == NULL) {

```

```

    cursorsize(-1,-1);
    writes(STARTX,ENDY-2,"Cost file doesn't exist, press any key ...",NORMAL);
    getch();
    clrline(ENDY-2,NORMAL);
    return;
}
for (i = 1;i <= MAXDIA;i++)
    fscanf(fp,"%lf",&Circ[i-1].cost);
for (i = 1;i <= MAXDIM;i++)
    fscanf(fp,"%lf",&Rect[i-1].cost);
fscanf(fp,"%lf",&Uexccost);
fscanf(fp,"%lf",&Ubkfcost);
fscanf(fp,"%lf",&Ushpcost);
fscanf(fp,"%lf",&Ulngcost);
fscanf(fp,"%lf",&Uprpcost);
fscanf(fp,"%lf",&Cover);
fscanf(fp,"%lf",&Freebrd);
fscanf(fp,"%lf",&Sslope);
fscanf(fp,"%lf",&Shpdepth);
fclose(fp);
Eleest = 0;
Totest = 0;
cursorsize(-1,-1);
}

void savecost(void)
{
    register int i;
    FILE *fp;
    char str[9];
    char fnamecst[13];
    char *cst = ".cst";

    cursorsize(13,14);
    writes(STARTX,ENDY-2,"Save cost to file :",NORMAL);
    gotoxy(STARTX+20,ENDY-2);
    scanf("%s",str);
    clrline(ENDY-2,NORMAL);
    strcpy(fnamecst,str);
    strcat(fnamecst,cst);
    fp = fopen(fnamecst,"w");
    for (i = 1;i <= MAXDIA;i++)
        fprintf(fp,"%11.2f\n",Circ[i-1].cost);
    for (i = 1;i <= MAXDIM;i++)
        fprintf(fp,"%11.2f\n",Rect[i-1].cost);
    fprintf(fp,"%11.2f\n",Uexccost);
    fprintf(fp,"%11.2f\n",Ubkfcost);
    fprintf(fp,"%11.2f\n",Ushpcost);
    fprintf(fp,"%11.2f\n",Ulngcost);
    fprintf(fp,"%11.2f\n",Uprpcost);
    fprintf(fp,"%4.2f\n",Cover);
    fprintf(fp,"%4.2f\n",Freebrd);
    fprintf(fp,"%3.1f\n",Sslope);
    fprintf(fp,"%4.2f\n",Shpdepth);
    fclose(fp);
    cursorsize(-1,-1);
}

void printcost(void)
{
    register int i;

    if (!Eleest || !Totest)
        return;
    fprintf(stdprn,"Circular Pipe Cost
    Rectangular Pipe Cost\n\r");
    fprintf(stdprn,"-----
    -----
    \n\n\r");
    for (i = 1;i <= MAXDIA;i++) {
        fprintf(stdprn,"%4.2f m. : %7.0f Baht/m.
        ",Circ[i-1].dia,Circ[i-1].cost);
        fprintf(stdprn,"
        %4.2f m. x %4.2f m. : %7.0f Baht/m.\n\r",Rect[i-1].w,
        Rect[i-1].h,Rect[i-1].cost);
    }
    fprintf(stdprn,"
    Rect[MAXDIM-1].w,Rect[MAXDIM-1].h,Rect[MAXDIM-1].cost);
    fprintf(stdprn,"Civil Work Cost\n\r");
    fprintf(stdprn,"-----
    \n\n\r");
    fprintf(stdprn,"Excavation cost
    : %7.0f Baht/cu.m.\n\r",Uexccost);
    fprintf(stdprn,"Backfill cost
    : %7.0f Baht/cu.m.\n\r",Ubkfcost);
    fprintf(stdprn,"Sheet pile cost
    : %7.0f Baht/m.\n\r",Ushpcost);
}

```



```

fprintf(stdprn,"Concrete lining cost      : %7.0f Baht/sq.m.\n\r",Ulngcost);
fprintf(stdprn,"Pavement repair cost      : %7.0f Baht/sq.m.\n\r",Uprpcost);
fprintf(stdprn,"Minimum cover depth      : %7.2f m.\n\r",Cover);
fprintf(stdprn,"Minimum freeboard      : %7.2f m.\n\r",Freebrd);
fprintf(stdprn,"Excavation side slope   : %7.1f\n\r",Sslope);
fprintf(stdprn,"Depth to use sheet pile : %7.2f m.\n\r",Shpdepth);
fprintf(stdprn,"Element Cost (Baht)\n\r");
fprintf(stdprn,"-----\n\r");
fprintf(stdprn," No.      Element      Excavate      Backfill      Sheet pile      Lining      Pavement\n\r");
for (i = 1;i <= Elenum;i++)
    fprintf(stdprn,"%5d %11.2f %11.2f %11.2f %11.2f %11.2f %11.2f\n\r",
        Eleno[i-1],Elecost[i-1],Exccost[i-1],Bkfcost[i-1],Shpcost[i-1],
        Lngcost[i-1],Prpcost[i-1]);
fprintf(stdprn,"Total Cost\n\r");
fprintf(stdprn,"-----\n\r");
fprintf(stdprn,"Total element cost      : %12.2f Baht.\n\r",Telecost);
fprintf(stdprn,"Total excavation cost   : %12.2f Baht.\n\r",Texccost);
fprintf(stdprn,"Total backfill cost     : %12.2f Baht.\n\r",Tbkfcost);
fprintf(stdprn,"Total sheet pile cost   : %12.2f Baht.\n\r",Tshpcost);
fprintf(stdprn,"Total concrete lining cost : %12.2f Baht.\n\r",Tlngcost);
fprintf(stdprn,"Total pavement repair cost : %12.2f Baht.\n\r",Tprpcost);
fprintf(stdprn,"Grand Total             : %12.2f Baht.\n\r",Totalcost);
}

void edit_circcost(void)
{
    register int i;
    int circ = 0;
    char block[4096];

    for (i = 1;i <= Elenum;i++) {
        if (Eletype[i-1] == 1)
            circ = 1;
    }
    if (!circ)
        return;
    gettext(STARTX,STARTY,ENDX,ENDY,block);
    clrscr();
    writes(MINX,MINY,Tagcost,REVERSE);
    writes(MINX,MAXY,Hlpcost2,REVERSE);
    cursize(13,14);
    writes(STARTX+30,STARTY+3," Circular Pipe Cost ",REVERSE);
    for (i = 1;i <= MAXDIA;i++) {
        gotoxy(STARTX+28,STARTY+i+6);
        cprintf("%4.2f m. : %7.0f Baht/m.",Circ[i-1].dia,Circ[i-1].cost);
    }
    for (i = 1;i <= MAXDIA;i++) {
        gotoxy(STARTX+38,STARTY+i+6);
        scanf("%7lf",&Circ[i-1].cost);
        gotoxy(STARTX+38,STARTY+i+6);
        cprintf("%7.0f",Circ[i-1].cost);
    }
    Eleest = 0;
    Totest = 0;
    puttext(STARTX,STARTY,ENDX,ENDY,block);
    cursize(-1,-1);
}

void edit_rectcost(void)
{
    register int i,j;
    int rect = 0;
    char block[4096];

    for (i = 1;i <= Elenum;i++) {
        if (Eletype[i-1] == 2)
            rect = 1;
    }
    if (!rect)
        return;
    gettext(STARTX,STARTY,ENDX,ENDY,block);
    clrscr();
    writes(MINX,MINY,Tagcost,REVERSE);
    writes(MINX,MAXY,Hlpcost2,REVERSE);
    cursize(13,14);
    writes(STARTX+28,STARTY+3," Rectangular Pipe Cost ",REVERSE);
    for (i = 1;i <= MAXDIM;i++) {
        gotoxy(STARTX+22,STARTY+i+6);

```

```

        cprintf("%4.2f m. x %4.2f m. : %7.0f Baht/m.",Rect[i-1].w,Rect[i-1].h,
                Rect[i-1].cost);
    }
    for (i = 1;i <= MAXDIM;i++) {
        gotoxy(STARTX+42,STARTY+i+6);
        scanf("%7lf",&Rect[i-1].cost);
        gotoxy(STARTX+42,STARTY+i+6);
        cprintf("%7.0f",Rect[i-1].cost);
    }
    Elecost = 0;
    Totest = 0;
    puttext(STARTX,STARTY,ENDX,ENDY,block);
    cursorsize(-1,-1);
}

void edit_civlcost(void)
{
    register int i;
    char block[4096];

    gettext(STARTX,STARTY,ENDX,ENDY,block);
    clrscr();
    writes(MINX,MINY,Tagcost,REVERSE);
    writes(MINX,MAXY,Hlpcost2,REVERSE);
    cursorsize(13,14);
    writes(STARTX+31,STARTY+3," Civil Work Cost ",REVERSE);
    gotoxy(STARTX+18,STARTY+7);
    cprintf("Excavation cost           : %7.0f Baht/cu.m.",Uexccost);
    gotoxy(STARTX+18,STARTY+8);
    cprintf("Backfill cost           : %7.0f Baht/cu.m.",Ubkfcost);
    gotoxy(STARTX+18,STARTY+9);
    cprintf("Sheet pile cost           : %7.0f Baht/m.",Ushpcost);
    gotoxy(STARTX+18,STARTY+10);
    cprintf("Concrete lining cost      : %7.0f Baht/sq.m.",Ulngcost);
    gotoxy(STARTX+18,STARTY+11);
    cprintf("Pavement repair cost     : %7.0f Baht/sq.m.",Uprpcost);
    gotoxy(STARTX+18,STARTY+12);
    cprintf("Minimum cover depth      : %7.2f m.",Cover);
    gotoxy(STARTX+18,STARTY+13);
    cprintf("Minimum freeboard        : %7.2f m.",Freebrd);
    gotoxy(STARTX+18,STARTY+14);
    cprintf("Excavation side slope    : %7.1f ",Sslope);
    gotoxy(STARTX+18,STARTY+15);
    cprintf("Depth to use sheet pile  : %7.2f m.",Shpdepth);
    gotoxy(STARTX+44,STARTY+7);
    scanf("%7lf",&Uexccost);
    gotoxy(STARTX+44,STARTY+7);
    cprintf("%7.0f",Uexccost);
    gotoxy(STARTX+44,STARTY+8);
    scanf("%7lf",&Ubkfcost);
    gotoxy(STARTX+44,STARTY+8);
    cprintf("%7.0f",Ubkfcost);
    gotoxy(STARTX+44,STARTY+9);
    scanf("%7lf",&Ushpcost);
    gotoxy(STARTX+44,STARTY+9);
    cprintf("%7.0f",Ushpcost);
    gotoxy(STARTX+44,STARTY+10);
    scanf("%7lf",&Ulngcost);
    gotoxy(STARTX+44,STARTY+10);
    cprintf("%7.0f",Ulngcost);
    gotoxy(STARTX+44,STARTY+11);
    scanf("%7lf",&Uprpcost);
    gotoxy(STARTX+44,STARTY+11);
    cprintf("%7.0f",Uprpcost);
    gotoxy(STARTX+44,STARTY+12);
    scanf("%7lf",&Cover);
    gotoxy(STARTX+44,STARTY+12);
    cprintf("%7.2f",Cover);
    gotoxy(STARTX+44,STARTY+13);
    scanf("%7lf",&Freebrd);
    gotoxy(STARTX+44,STARTY+13);
    cprintf("%7.2f",Freebrd);
    gotoxy(STARTX+44,STARTY+14);
    scanf("%7lf",&Sslope);
    gotoxy(STARTX+44,STARTY+14);
    cprintf("%7.1f",Sslope);
    gotoxy(STARTX+44,STARTY+15);
    scanf("%7lf",&Shpdepth);
}

```

```

gotoxy(STARTX+44,STARTY+15);
cprintf("%7.2f",Shpdepth);
Eleest = 0;
Tolest = 0;
puttext(STARTX,STARTY,ENDX,ENDY,block);
cursormsize(-1,-1);
}

void elementcost(void)
{
register int i;
union key {
char ch[2];
unsigned i;
} c;
char block[4096];

getttext(STARTX,STARTY,ENDX,ENDY,block);
clrscr();
writes(MINX,MINY,Tagcost,REVERSE);
for (i = 1;i <= Elenum;i++) {
switch (Eletype[i-1]) {
case 1 : circcost(i);
break;
case 2 : rectcost(i);
break;
case 13 : trapcost(i);
}
}
Eleest = 1;
writes(STARTX+28,STARTY+2," Element Cost (Baht) ",REVERSE);
writes(STARTX+2,STARTY+5,"No. Element Excavate Backfill Sheet pile Lining Paveme:
NORMAL);
for (i = 1;i <= 13;i++) {
gotoxy(STARTX,STARTY+i+6);
cprintf("%5d %11.2f %11.2f %11.2f %11.2f %11.2f %11.2f",
Eleno[i-1],Elecost[i-1],Exccost[i-1],Bkfcost[i-1],Shpcost[i-1],
Lngcost[i-1],Prpcost[i-1]);
}
for (;;) {
writes(MINX,MAXY,Hlpcost4,REVERSE);
while (!bioskey(1));
c.i = bioskey(0);
if (!c.ch[0]) {
switch (c.ch[1]) {
case UP : if (i > 13) {
scrollldn(STARTX,STARTY+7,ENDX,STARTY+19,1,NORMAL);
gotoxy(STARTX,STARTY+7);
cprintf("%5d %11.2f %11.2f %11.2f %11.2f %11.2f %11.2f",
Eleno[i-14],Elecost[i-14],Exccost[i-14],Bkfcost[i-14],
Shpcost[i-14],Lngcost[i-14],Prpcost[i-14]);
i--;
}
break;
case DN : if (i < Elenum) {
scrollup(STARTX,STARTY+7,ENDX,STARTY+19,1,NORMAL);
gotoxy(STARTX,STARTY+19);
cprintf("%5d %11.2f %11.2f %11.2f %11.2f %11.2f %11.2f",
Eleno[i],Elecost[i],Exccost[i],Bkfcost[i],
Shpcost[i],Lngcost[i],Prpcost[i]);
i++;
}
break;
case F4 : puttext(STARTX,STARTY,ENDX,ENDY,block);
return;
case F10 : totalcost();
break;
}
}
}
}

void totalcost(void)
{
register int i;
char block[4096];

if (!Eleest)

```

```

return;
getttext(STARTX,STARTY,ENDX,ENDY,block);
clrscr();
writes(MINX,MINY,Tagcost,REVERSE);
Telecost = Texccost = Tbkfcost = Tshpcost = Tlngcost = Tprpcost = 0;
writes(STARTX+32,STARTY+3," Total Cost ",REVERSE);
writes(STARTX+15,STARTY+7,"Total element cost      :      Baht.",NORMAL);
writes(STARTX+15,STARTY+8,"Total excavation cost      :      Baht.",NORMAL);
writes(STARTX+15,STARTY+9,"Total backfill cost      :      Baht.",NORMAL);
writes(STARTX+15,STARTY+10,"Total sheet pile cost     :      Baht.",NORMAL);
writes(STARTX+15,STARTY+11,"Total concrete lining cost :      Baht.",NORMAL);
writes(STARTX+15,STARTY+12,"Total pavement repair cost :      Baht.",NORMAL);
for (i = 1;i <= Elenum;i++) {
    Telecost += Elecost[i-1];
    Texccost += Exccost[i-1];
    Tbkfcost += Bkfcost[i-1];
    Tshpcost += Shpcost[i-1];
    Tlngcost += Lngcost[i-1];
    Tprpcost += Prpcost[i-1];
}
gotoxy(STARTX+44,STARTY+7);
cprintf("%12.2f",Telecost);
gotoxy(STARTX+44,STARTY+8);
cprintf("%12.2f",Texccost);
gotoxy(STARTX+44,STARTY+9);
cprintf("%12.2f",Tbkfcost);
gotoxy(STARTX+44,STARTY+10);
cprintf("%12.2f",Tshpcost);
gotoxy(STARTX+44,STARTY+11);
cprintf("%12.2f",Tlngcost);
gotoxy(STARTX+44,STARTY+12);
cprintf("%12.2f",Tprpcost);
Totalcost = Telecost + Texccost + Tbkfcost + Tshpcost + Tlngcost + Tprpcost;
gotoxy(STARTX+15,STARTY+14);
cprintf("Grand Total          %12.2f Baht.",Totalcost);
writes(MINX,MAXY,Hlpcost3,REVERSE);
Totest = 1;
getch();
putttext(STARTX,STARTY,ENDX,ENDY,block);
}

void circcost(int i)
{
    register int k;
    double depth,width,bwidth,sectarea,circarea;

    for (k = 1;k <= MAXDIA;k++) {
        if (Elegeo1[i-1] == Circ[k-1].dia) {
            Elecost[i-1] = (Elelen[i-1] * Circ[k-1].cost);
            depth      = Circ[k-1].dia + Cover;
            bwidth     = Circ[k-1].dia + 1;
            if (depth >= Shpdepth) {
                width      = bwidth;
                Shpcost[i-1] = (Elelen[i-1] * Ushpcost);
            }
            else
                width = bwidth + 2*Sslope*depth;
            sectarea   = 0.5 * depth * (width+bwidth);
            circarea   = PI * pow(Circ[k-1].dia,2) / 4;
            Exccost[i-1] = (sectarea * Elelen[i-1] * Uexccost);
            Bkfcost[i-1] = ((sectarea - circarea) * Elelen[i-1] * Ubkfcost);
            Prpcost[i-1] = (width * Elelen[i-1] * Uprpcost);
        }
    }
}

void rectcost(int i)
{
    register int k;
    double depth,width,bwidth,sectarea,rectarea;

    for (k = 1;k <= MAXDIM;k++) {
        if ((Elegeo1[i-1]==Rect[k-1].h) && (Elegeo2[i-1]==Rect[k-1].w)) {
            Elecost[i-1] = (Elelen[i-1] * Rect[k-1].cost);
            depth      = Rect[k-1].h + Cover;
            bwidth     = Rect[k-1].w + 1;
            if (depth >= Shpdepth) {
                width      = bwidth;
            }
        }
    }
}

```

```

    Shpcost[i-1] = (Elelen[i-1] * Ushpcost);
}
else
width      = bwidth + 2*Sslope*depth;
sectarea   = 0.5 * depth * (width+bwidth);
rectarea   = Rect[k-1].w * Rect[k-1].h;
Exccost[i-1] = (sectarea * Elelen[i-1] * Uexccost);
Bkfcost[i-1] = ((sectarea - rectarea) * Elelen[i-1] * Ubkfcost);
Prpcost[i-1] = (width * Elelen[i-1] * Uprpcost);
}
}
}

void trapcost(int i)
{
double side,wp,depth,width,sectarea;

side      = pow(pow(Elegeo1[i-1],2)+pow(Elegeo1[i-1]*Elegeo3[i-1],2),0.5);
wp        = Elegeo2[i-1] + 2*side;
Lngcost[i-1] = wp * Elelen[i-1] * Ulngcost;
depth     = Elegeo1[i-1] + Freebrd;;
width     = Elegeo2[i-1] + 2*depth*Elegeo3[i-1];
sectarea  = 0.5 * depth * (width+Elegeo2[i-1]);
Exccost[i-1] = (sectarea * Elelen[i-1] * Uexccost);
}

void cursorsize(int startline,int stopline)
{
union REGS r;

r.h.ah = 1;
r.h.ch = startline;
r.h.cl = stopline;
int86(0x10,&r,&r);
}

void writech(int x,int y,unsigned char ch,int attr)
{
char far *m;

m = Vdomem + ((y-1)*160) + (x-1)*2;
*m++ = ch;
*m = attr;
}

void writes(int x,int y,unsigned char *s,int attr)
{
register char i;
char far *m;

m = Vdomem + ((y-1)*160) + (x-1)*2;
for (i = x; *s ;i++) {
*m++ = *s++;
*m++ = attr;
}
}

void clrline(int line,int attr)
{
register int x;

for (x = STARTX;x <= ENDX;x++)
writech(x,line,' ',attr);
}

void scrollup(int left,int top,int right,int bottom,int lineno,int attr)
{
union REGS r;

r.h.ah = 6;
r.h.al = lineno;
r.h.cl = left - 1;
r.h.ch = top - 1;
r.h.dl = right - 1;
r.h.dh = bottom - 1;
r.h.bh = attr;
int86(0x10,&r,&r);
}

```

```
void scrolldn(int left,int top,int right,int bottom,int lineno,int attr)
{
    union REGS r;

    r.h.ah = 7;
    r.h.al = lineno;
    r.h.cl = left - 1;
    r.h.ch = top - 1;
    r.h.dl = right - 1;
    r.h.dh = bottom - 1;
    r.h.bh = attr;
    int86(0x10,&r,&r);
}
```

ภาคผนวก ค.

รายละเอียดการประเมินราคาข้อมูลทดสอบ

## Circular Pipe Cost

0.30 m. :	215 Baht/m.
0.40 m. :	270 Baht/m.
0.50 m. :	325 Baht/m.
0.60 m. :	390 Baht/m.
0.80 m. :	650 Baht/m.
1.00 m. :	1030 Baht/m.
1.20 m. :	1400 Baht/m.
1.50 m. :	2070 Baht/m.
1.75 m. :	2900 Baht/m.
2.00 m. :	3980 Baht/m.
2.25 m. :	5350 Baht/m.
2.50 m. :	7100 Baht/m.

## Rectangular Pipe Cost

1.20 m. x 1.20 m. :	3150 Baht/m.
1.50 m. x 1.20 m. :	4300 Baht/m.
1.50 m. x 1.50 m. :	4800 Baht/m.
1.80 m. x 1.50 m. :	5450 Baht/m.
1.80 m. x 1.80 m. :	6230 Baht/m.
2.10 m. x 1.80 m. :	7390 Baht/m.
2.10 m. x 2.10 m. :	8600 Baht/m.
2.40 m. x 2.10 m. :	10100 Baht/m.
2.40 m. x 2.40 m. :	11700 Baht/m.
2.70 m. x 2.40 m. :	13500 Baht/m.
2.70 m. x 2.70 m. :	15500 Baht/m.
3.00 m. x 2.70 m. :	17600 Baht/m.
3.00 m. x 3.00 m. :	19800 Baht/m.

## Civil Work Cost

Excavation cost :	45 Baht/cu.m.
Backfill cost :	60 Baht/cu.m.
Sheet pile cost :	1100 Baht/m.
Concrete lining cost :	400 Baht/sq.m.
Pavement repair cost :	400 Baht/sq.m.
Minimum cover depth :	0.60 m.
Minimum freeboard :	0.30 m.
Excavation side slope :	1.0
Depth to use sheet pile :	1.50 m.

## Element Cost (Baht)

No.	Element	Excavate	Backfill	Sheet pile	Lining	Pavement
101	21500.00	8910.00	11455.89	0.00	0.00	124000.00
102	27000.00	10800.00	13646.02	0.00	0.00	136000.00
103	32500.00	12870.00	15981.90	0.00	0.00	148000.00
104	39000.00	15120.00	18463.54	0.00	0.00	160000.00
105	65000.00	20160.00	23864.07	0.00	0.00	184000.00
106	103000.00	14400.00	14487.61	110000.00	0.00	80000.00
107	140000.00	17820.00	16974.16	110000.00	0.00	88000.00
108	207000.00	23625.00	20897.13	110000.00	0.00	100000.00
109	290000.00	29081.25	24343.32	110000.00	0.00	110000.00
110	398000.00	35100.00	27950.46	110000.00	0.00	120000.00
111	535000.00	41681.25	31718.55	110000.00	0.00	130000.00
112	710000.00	48825.00	35647.59	110000.00	0.00	140000.00
113	0.00	0.00	0.00	0.00	0.00	0.00
201	315000.00	17820.00	15120.00	110000.00	0.00	88000.00
202	430000.00	20250.00	16200.00	110000.00	0.00	100000.00
203	480000.00	23625.00	18000.00	110000.00	0.00	100000.00
204	545000.00	26460.00	19080.00	110000.00	0.00	112000.00
205	623000.00	30240.00	20880.00	110000.00	0.00	112000.00
206	739000.00	33480.00	21960.00	110000.00	0.00	124000.00
207	860000.00	37665.00	23760.00	110000.00	0.00	124000.00
208	1010000.00	41310.00	24840.00	110000.00	0.00	136000.00
209	1170000.00	45900.00	26640.00	110000.00	0.00	136000.00
210	1350000.00	49950.00	27720.00	110000.00	0.00	148000.00
211	1550000.00	54945.00	29520.00	110000.00	0.00	148000.00
212	1760000.00	59400.00	30600.00	110000.00	0.00	160000.00
213	1980000.00	64800.00	32400.00	110000.00	0.00	160000.00
214	0.00	0.00	0.00	0.00	0.00	0.00
301	0.00	13455.00	0.00	0.00	153137.08	0.00
302	0.00	25155.00	0.00	0.00	233137.08	0.00
303	0.00	36855.00	0.00	0.00	313137.08	0.00
401	0.00	0.00	0.00	0.00	0.00	0.00

## Total Cost

Total element cost :	15380000.00 Baht.
Total excavation cost :	859702.50 Baht.
Total backfill cost :	562150.24 Baht.
Total sheet pile cost :	2200000.00 Baht.
Total concrete lining cost :	699411.25 Baht.
Total pavement repair cost :	3168000.00 Baht.

Grand Total 22869263.99 Baht.



ภาคผนวก ง.

ข้อมูลเข้าและผลลัพธ์จากกรณีศึกษา

```

*****
**** Executive Block
SW 2 0 8 8 0
MM 2 1 2
@ 8 'ASGAME.INT'
*****
**** RUNOFF Block
$RUNOFF
A1 '13th ASIAN GAMES, RANGSIT CAMPUS THAMMASAT UNIVERSITY.'
A1 ''
* METRIC ISNOW NRGAG INFILM KWALTY IVAP NHR NMN NDAY MONTH IYRSTR
B1 1 0 1 0 0 0 0 0 1 4 96
* IPRN(1) IPRN(2) IPRN(3)
B2 0 1 0
* WET WETDRY DRY LUNIT LONG
B3 300 600 3600 1 180
D1 0
**** Rainfall data
* KTYPE KINC KPRINT KTHIS KTIME KPREP NHISTO THISTO TZRAIN
E1 0 10 1 0 0 0 18 10 0
E3 15.72 44.42 159.36 102.60 71.71 53.05 40.89 32.53 26.53 22.07
E3 18.67 16.01 13.90 12.18 10.78 9.61 8.62 7.79
**** Catchment data
* JK NAMEW NGTO WIDTH WAREA WW3 WW4 WW5 WW6 WW7 WW8 WW9 WW10 WW11
H1 1 1 9421 350 6.14 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 2 9401 345 9.80 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 3 9401 345 7.52 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 4 9411 450 6.81 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 5 9412 540 3.83 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 6 9431 530 5.49 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 7 9441 265 7.45 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 8 9441 265 5.94 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 9 9404 450 5.05 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 10 9451 570 7.74 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 11 9452 170 3.33 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 12 9452 170 3.17 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 13 9452 170 3.79 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 14 9405 520 5.81 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 15 9461 350 4.44 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 16 9405 520 3.37 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 17 9106 335 8.35 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 18 9106 335 4.92 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 19 9301 690 10.55 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 20 9311 345 8.45 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 21 9311 345 8.92 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 22 9321 340 2.72 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 23 9321 340 4.65 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 24 9102 210 3.20 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 25 9331 225 2.26 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 26 9331 225 3.22 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 27 9331 225 3.32 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 28 9304 220 3.48 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 29 9341 335 3.11 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 30 9341 335 3.29 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 31 9351 335 5.41 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 32 9351 335 5.42 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 33 9304 220 1.88 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 34 9304 220 5.64 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 35 9104 105 4.73 75 .0001 .02 .1 5 20 20 2.5 .001
H1 1 36 9104 105 4.91 75 .0001 .02 .1 5 20 20 2.5 .001
H1 1 37 9205 560 4.24 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 38 9101 390 7.74 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 39 9101 390 6.26 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 40 9201 370 2.97 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 41 9111 345 6.21 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 42 9111 345 6.89 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 43 9201 210 2.46 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 44 9121 690 5.60 80 .0001 .02 .1 5 20 20 2.5 .001
H1 1 45 9131 690 5.90 80 .0001 .02 .1 5 20 20 2.5 .001
H1 1 46 9212 210 2.34 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 47 9202 310 2.58 50 .0001 .02 .1 5 20 20 2.5 .001
H1 1 48 9141 230 5.31 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 49 9141 230 4.76 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 50 9141 230 2.39 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 51 9214 360 4.82 60 .0001 .02 .1 5 20 20 2.5 .001
H1 1 52 9214 360 4.21 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 53 9203 310 3.68 30 .0001 .02 .1 5 20 20 2.5 .001
H1 1 54 9204 375 3.26 60 .0001 .02 .1 5 20 20 2.5 .001

```

H1 1 55 9204 375 3.25 60 .0001 .02 .1 5 20 20 2.5 .001  
M1 0 0

\*\*\*\*\*  
\*\*\*\* TRANSPORT Block

\$TRANSPORT

A1 '13th ASIAN GAMES, RANGSIT CAMPUS THAMMASAT UNIVERSITY.'

A1 ''  
\* NDT NINPUT NNYN NNPE NOUTS NPRINT NPOLL NITER IDATEZ METRIC INTPRT  
B1 40 0 0 0 0 1 0 4 960401 1 10

\* DT EPSIL DWDAYS TZERO GNU TRIBA  
B2 300 0.0001 0 0.0 0.01 242.33

\* NCNTRL NINFIL NFILTH NDESN

B3 0 0 0 1

\* NKLASS KPRINT

C1 0 0

\*\*\*\* Channel data

* NOE	NUE(1)	NUE(2)	NUE(3)	NTYPE	DIST	GEOM1	SLOPE	ROUGH	GEOM2	BARREL	GEOM3
E1 9101	0	0	0	19	0	0	0	0	0	1	0
E1 9102	101	0	0	19	0	0	0	0	0	1	0
E1 9103	102	111	121	19	0	0	0	0	0	1	0
E1 9104	103	131	141	19	0	0	0	0	0	1	0
E1 9105	104	205	304	19	0	0	0	0	0	1	0
E1 9106	105	0	0	19	0	0	0	0	0	1	0
E1 9111	0	0	0	19	0	0	0	0	0	1	0
E1 9121	0	0	0	19	0	0	0	0	0	1	0
E1 9131	0	0	0	19	0	0	0	0	0	1	0
E1 9141	0	0	0	19	0	0	0	0	0	1	0
E1 9201	0	0	0	19	0	0	0	0	0	1	0
E1 9202	201	0	0	19	0	0	0	0	0	1	0
E1 9203	202	0	0	19	0	0	0	0	0	1	0
E1 9204	203	0	0	19	0	0	0	0	0	1	0
E1 9205	204	214	0	19	0	0	0	0	0	1	0
E1 9211	0	0	0	19	0	0	0	0	0	1	0
E1 9212	211	0	0	19	0	0	0	0	0	1	0
E1 9213	212	0	0	19	0	0	0	0	0	1	0
E1 9214	213	0	0	19	0	0	0	0	0	1	0
E1 9301	0	0	0	19	0	0	0	0	0	1	0
E1 9302	301	311	0	19	0	0	0	0	0	1	0
E1 9303	302	321	331	19	0	0	0	0	0	1	0
E1 9304	303	341	351	19	0	0	0	0	0	1	0
E1 9311	0	0	0	19	0	0	0	0	0	1	0
E1 9321	0	0	0	19	0	0	0	0	0	1	0
E1 9331	0	0	0	19	0	0	0	0	0	1	0
E1 9341	0	0	0	19	0	0	0	0	0	1	0
E1 9351	0	0	0	19	0	0	0	0	0	1	0
E1 9401	0	0	0	19	0	0	0	0	0	1	0
E1 9402	401	0	0	19	0	0	0	0	0	1	0
E1 7403	421	431	8403	19	0	0	0	0	0	1	0
E1 9403	402	412	0	19	0	0	0	0	0	1	0
E1 9404	403	441	0	19	0	0	0	0	0	1	0
E1 9405	404	452	0	19	0	0	0	0	0	1	0
E1 9411	0	0	0	19	0	0	0	0	0	1	0
E1 9412	411	0	0	19	0	0	0	0	0	1	0
E1 9421	0	0	0	19	0	0	0	0	0	1	0
E1 9431	0	0	0	19	0	0	0	0	0	1	0
E1 9441	0	0	0	19	0	0	0	0	0	1	0
E1 9451	0	0	0	19	0	0	0	0	0	1	0
E1 9452	451	461	0	19	0	0	0	0	0	1	0
E1 9461	0	0	0	19	0	0	0	0	0	1	0
E1 101	9101	0	0	13	780	1.30	.020	.013	3.00	1	1
E1 102	9102	0	0	13	370	1.40	.020	.013	3.00	1	1
E1 103	9103	0	0	13	210	1.40	.020	.013	3.00	1	1
E1 104	9104	0	0	13	810	1.50	.020	.013	5.00	1	1
E1 105	9105	0	0	13	210	1.80	.033	.013	7.00	1	1
E1 106	9106	0	0	2	670	3.00	.100	.013	3.00	1	0
E1 111	9111	0	0	13	690	1.00	.020	.013	3.00	1	1
E1 121	9121	0	0	13	690	1.00	.033	.013	1.00	1	1
E1 131	9131	0	0	13	690	1.00	.033	.013	1.00	1	1
E1 141	9141	0	0	13	690	1.30	.033	.013	3.00	1	1
E1 201	9201	0	0	13	370	1.00	.100	.013	3.00	1	1
E1 202	9202	0	0	13	210	1.00	.033	.013	3.00	1	1
E1 203	9203	0	0	13	310	1.20	.033	.013	3.00	1	1
E1 204	9204	0	0	13	750	1.30	.033	.013	3.00	1	1
E1 205	9205	0	0	13	560	1.60	.033	.013	3.00	1	1
E1 211	9211	0	0	13	370	0.80	.033	.013	1.00	1	1
E1 212	9212	0	0	13	210	0.80	.033	.013	1.00	1	1
E1 213	9213	0	0	13	420	1.00	.033	.013	1.00	1	1
E1 214	9214	0	0	13	720	1.20	.033	.013	3.00	1	1

E1	301	9301	0	0	13	690	1.20	.033	.013	3.00	1	1
E1	302	9302	0	0	13	350	1.50	.033	.013	5.00	1	1
E1	303	9303	0	0	13	200	1.60	.033	.013	5.00	1	1
E1	304	9304	0	0	13	650	1.80	.033	.013	5.00	1	1
E1	311	9311	0	0	13	690	1.40	.033	.013	3.00	1	1
E1	321	9321	0	0	13	680	1.20	.033	.013	3.00	1	1
E1	331	9331	0	0	13	680	1.30	.033	.013	3.00	1	1
E1	341	9341	0	0	13	670	1.40	.033	.013	3.00	1	1
E1	351	9351	0	0	13	670	1.20	.033	.013	3.00	1	1
E1	401	9401	0	0	13	690	1.50	.033	.013	3.00	1	1
E1	402	9402	0	0	2	400	2.10	.100	.013	2.40	1	0
E1	403	7403	0	0	13	210	1.50	.033	.013	5.00	1	1
E1	8403	9403	0	0	2	25	3.00	.100	.013	3.00	1	0
E1	404	9404	0	0	13	450	1.70	.033	.013	5.00	1	1
E1	405	9405	0	0	2	520	3.00	.100	.013	3.00	1	0
E1	411	9411	0	0	13	450	1.00	.033	.013	3.00	1	1
E1	412	9412	0	0	2	540	2.10	.100	.013	2.40	1	0
E1	421	9421	0	0	2	350	2.10	.100	.013	2.40	1	0
E1	431	9431	0	0	2	530	2.10	.100	.013	2.40	1	0
E1	441	9441	0	0	13	530	1.00	.033	.013	3.00	1	1
E1	451	9451	0	0	13	570	0.80	.033	.013	3.00	1	1
E1	452	9452	0	0	13	500	1.00	.033	.013	3.00	1	1
E1	461	9461	0	0	13	350	0.80	.033	.013	3.00	1	1

\$ENDPROGRAM

## Circular Pipe Cost

0.30 m. :	215 Baht/m.
0.40 m. :	270 Baht/m.
0.50 m. :	325 Baht/m.
0.60 m. :	390 Baht/m.
0.80 m. :	650 Baht/m.
1.00 m. :	1030 Baht/m.
1.20 m. :	1400 Baht/m.
1.50 m. :	2070 Baht/m.
1.75 m. :	2900 Baht/m.
2.00 m. :	3980 Baht/m.
2.25 m. :	5350 Baht/m.
2.50 m. :	7100 Baht/m.

## Rectangular Pipe Cost

1.20 m. x 1.20 m. :	2835 Baht/m.
1.50 m. x 1.20 m. :	3870 Baht/m.
1.50 m. x 1.50 m. :	4320 Baht/m.
1.80 m. x 1.50 m. :	4905 Baht/m.
1.80 m. x 1.80 m. :	5605 Baht/m.
2.10 m. x 1.80 m. :	6650 Baht/m.
2.10 m. x 2.10 m. :	7740 Baht/m.
2.40 m. x 2.10 m. :	9090 Baht/m.
2.40 m. x 2.40 m. :	10530 Baht/m.
2.70 m. x 2.40 m. :	12150 Baht/m.
2.70 m. x 2.70 m. :	13950 Baht/m.
3.00 m. x 2.70 m. :	15840 Baht/m.
3.00 m. x 3.00 m. :	17820 Baht/m.

## Civil Work Cost

Excavation cost :	45 Baht/cu.m.
Backfill cost :	60 Baht/cu.m.
Sheet pile cost :	1100 Baht/m.
Concrete lining cost :	400 Baht/sq.m.
Pavement repair cost :	0 Baht/sq.m.
Minimum cover depth :	0.60 m.
Minimum freeboard :	0.20 m.
Excavation side slope :	1.0
Depth to use sheet pile :	1.50 m.

## Element Cost (Baht)

No.	Element	Excavate	Backfill	Sheet pile	Lining	Pavement
9101	0.00	0.00	0.00	0.00	0.00	0.00
9102	0.00	0.00	0.00	0.00	0.00	0.00
9103	0.00	0.00	0.00	0.00	0.00	0.00
9104	0.00	0.00	0.00	0.00	0.00	0.00
9105	0.00	0.00	0.00	0.00	0.00	0.00
9106	0.00	0.00	0.00	0.00	0.00	0.00
9111	0.00	0.00	0.00	0.00	0.00	0.00
9121	0.00	0.00	0.00	0.00	0.00	0.00
9131	0.00	0.00	0.00	0.00	0.00	0.00
9141	0.00	0.00	0.00	0.00	0.00	0.00
9201	0.00	0.00	0.00	0.00	0.00	0.00
9202	0.00	0.00	0.00	0.00	0.00	0.00
9203	0.00	0.00	0.00	0.00	0.00	0.00
9204	0.00	0.00	0.00	0.00	0.00	0.00
9205	0.00	0.00	0.00	0.00	0.00	0.00
9211	0.00	0.00	0.00	0.00	0.00	0.00
9212	0.00	0.00	0.00	0.00	0.00	0.00
9213	0.00	0.00	0.00	0.00	0.00	0.00
9214	0.00	0.00	0.00	0.00	0.00	0.00
9301	0.00	0.00	0.00	0.00	0.00	0.00
9302	0.00	0.00	0.00	0.00	0.00	0.00
9303	0.00	0.00	0.00	0.00	0.00	0.00
9304	0.00	0.00	0.00	0.00	0.00	0.00
9311	0.00	0.00	0.00	0.00	0.00	0.00
9321	0.00	0.00	0.00	0.00	0.00	0.00
9331	0.00	0.00	0.00	0.00	0.00	0.00
9341	0.00	0.00	0.00	0.00	0.00	0.00
9351	0.00	0.00	0.00	0.00	0.00	0.00
9401	0.00	0.00	0.00	0.00	0.00	0.00
9402	0.00	0.00	0.00	0.00	0.00	0.00
7403	0.00	0.00	0.00	0.00	0.00	0.00
9403	0.00	0.00	0.00	0.00	0.00	0.00
9404	0.00	0.00	0.00	0.00	0.00	0.00
9405	0.00	0.00	0.00	0.00	0.00	0.00
9411	0.00	0.00	0.00	0.00	0.00	0.00
9412	0.00	0.00	0.00	0.00	0.00	0.00
9421	0.00	0.00	0.00	0.00	0.00	0.00
9431	0.00	0.00	0.00	0.00	0.00	0.00
9441	0.00	0.00	0.00	0.00	0.00	0.00
9451	0.00	0.00	0.00	0.00	0.00	0.00
9452	0.00	0.00	0.00	0.00	0.00	0.00
9461	0.00	0.00	0.00	0.00	0.00	0.00
101	0.00	236925.00	0.00	0.00	2083210.04	0.00
102	0.00	122544.00	0.00	0.00	1030050.10	0.00
103	0.00	69552.00	0.00	0.00	584623.03	0.00
104	0.00	415165.50	0.00	0.00	2994615.58	0.00
105	0.00	170100.00	0.00	0.00	1015658.18	0.00
106	11939400.00	434160.00	217080.00	737000.00	0.00	0.00

111	0.00	156492.00	0.00	0.00	1608645.89	0.00
121	0.00	81972.00	0.00	0.00	1056645.89	0.00
131	0.00	81972.00	0.00	0.00	1056645.89	0.00
141	0.00	209587.50	0.00	0.00	1842839.65	0.00
201	0.00	83916.00	0.00	0.00	862607.21	0.00
202	0.00	47628.00	0.00	0.00	489587.88	0.00
203	0.00	85932.00	0.00	0.00	792869.96	0.00
204	0.00	227812.50	0.00	0.00	2003086.58	0.00
205	0.00	217728.00	0.00	0.00	1685708.28	0.00
211	0.00	33300.00	0.00	0.00	482885.77	0.00
212	0.00	18900.00	0.00	0.00	274070.30	0.00
213	0.00	49896.00	0.00	0.00	643175.76	0.00
214	0.00	199584.00	0.00	0.00	1841504.41	0.00
301	0.00	191268.00	0.00	0.00	1764775.06	0.00
302	0.00	179392.50	0.00	0.00	1293969.70	0.00
303	0.00	110160.00	0.00	0.00	762038.67	0.00
304	0.00	409500.00	0.00	0.00	2623703.89	0.00
311	0.00	228528.00	0.00	0.00	1920904.24	0.00
321	0.00	188496.00	0.00	0.00	1739198.61	0.00
331	0.00	206550.00	0.00	0.00	1816131.83	0.00
341	0.00	221904.00	0.00	0.00	1865225.86	0.00
351	0.00	185724.00	0.00	0.00	1713622.16	0.00
401	0.00	248089.50	0.00	0.00	1998968.83	0.00
402	3636000.00	165240.00	99360.00	440000.00	0.00	0.00
403	0.00	107635.50	0.00	0.00	776381.82	0.00
8403	445500.00	16200.00	8100.00	27500.00	0.00	0.00
404	0.00	265477.50	0.00	0.00	1765498.70	0.00
405	9266400.00	336960.00	168480.00	572000.00	0.00	0.00
411	0.00	102060.00	0.00	0.00	1049116.88	0.00
412	4908600.00	223074.00	134136.00	594000.00	0.00	0.00
421	3181500.00	144585.00	86940.00	385000.00	0.00	0.00
431	4817700.00	218943.00	131652.00	583000.00	0.00	0.00
441	0.00	120204.00	0.00	0.00	1235626.55	0.00
451	0.00	102600.00	0.00	0.00	1199905.11	0.00
452	0.00	113400.00	0.00	0.00	1165685.42	0.00
461	0.00	63000.00	0.00	0.00	736783.84	0.00

## Total Cost

-----

Total element cost	:	38195100.00	Baht.
Total excavation cost	:	7092157.50	Baht.
Total backfill cost	:	845748.00	Baht.
Total sheet pile cost	:	3338500.00	Baht.
Total concrete lining cost	:	47775967.59	Baht.
Total pavement repair cost	:	0.00	Baht.

Grand Total 97247473.09 Baht.



### รายการอ้างอิง

1. Wayne C. Huber and Robert E. Dickinson. Storm Water Management Model Version 4: User's Manual. U.S.Environmental Protection Agency. Athens. Georgia. 1988
2. American Society of Civil Engineers and Water Environment Systems. Design and Construction of Urban Stormwater Management System. WEF. ASCE. 1994.
3. Ven Te Chow, David R. Maidment and Larry W.Mays. Applied Hydrology. Singapore : Mc Graw-Hill. 1988
4. Borland International Inc.. Turbo C++ User's Guide. Borland International Inc. 1988



### ประวัติผู้เขียน

นายกิตติพล โสมสิริพัฒน์ เกิดเมื่อวันที่ 15 สิงหาคม 2512 สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสิ่งแวดล้อม จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2532 และสมัครเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมสิ่งแวดล้อม จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535 ช่วงเวลาหลังจากสำเร็จ การศึกษาปริญญาตรีจนถึงปัจจุบันทำงานอยู่ในบริษัท โปเรสเทคโนโลยีคอนซัลแต้นส์ จำกัด ใน ตำแหน่งวิศวกรออกแบบ