

วิชา การพัฒนาโปรแกรม (ภาษาซี)

รหัสวิชา ว31293

หน่วยการเรียนรู้ที่ 10 การจัดการไฟล์

สาระการเรียนรู้

1. ประเภทของไฟล์
2. การทำงานกับเท็กซ์ไฟล์
3. การใช้คำสั่งทำงานข้อมูลกับเท็กซ์ไฟล์

ผลการเรียนรู้

มีความรู้ความเข้าใจในการทำงานประมวลผลข้อมูลร่วมกับไฟล์ข้อมูลชนิดเท็กซ์ไฟล์และไฟล์แบบไบนารีได้นำไปประยุกต์ใช้งานในการเขียนโปรแกรมภาษาซีได้

จุดประสงค์การเรียนรู้

1. อธิบายความหมายของไฟล์ได้
2. อธิบายขบวนการประมวลผลกับไฟล์ได้
3. เข้าใจเกี่ยวกับไฟล์แบบเท็กซ์ไฟล์และไฟล์แบบไบนารีไฟล์
4. สามารถเขียนโปรแกรมเปิดและปิดไฟล์ได้
5. สามารถเขียนโปรแกรมเพื่อเก็บข้อมูลเป็นไฟล์ได้



ใบความรู้ที่ 10 เรื่อง การจัดการไฟล์

จัดทำโดย นางพรพนารัตน์ ชมภูนุช

ในงานบางประเภท จะต้องนำผลการทำงานของโปรแกรมเก็บลงในหน่วยความจำสำรอง ลักษณะของข้อมูลที่เก็บในหน่วยความจำสำรองนี้จะเก็บเป็นไฟล์ ซึ่งไฟล์ในภาษาซีมีอยู่หลายประเภท แต่ละประเภทมีคำสั่งในการอ่านเขียนไฟล์ต่างกัน ดังนั้นถ้าหากต้องการเขียนข้อมูลเป็นไฟล์จะต้องทำความเข้าใจฟังก์ชันที่กระทำกับไฟล์ด้วย



ประเภทของไฟล์

ในบทนี้จะกล่าวถึงลักษณะของไฟล์หรือแฟ้มข้อมูล และการเขียนโปรแกรมและอ่านไฟล์ กับหน่วยความจำสำรองที่ใช้เก็บข้อมูลเวลาปิดเครื่องคอมพิวเตอร์ หน่วยความจำนี้ข้อมูลจะไม่หาย เช่น ฮาร์ดไดรฟ์ หรือซีดีรอม ก่อนที่จะศึกษาเรื่องไฟล์จะต้องเข้าใจคำว่า สตรีม (stream) เสียก่อน



Stream คือ....

สตรีม (stream) เป็นหน่วยของข้อมูลที่เรียงติดกัน โดยข้อมูลแต่ละหน่วยอาจเป็นโครงสร้างหรือข้อมูลชนิดเดียวกันก็ได้ ส่วนไฟล์จะเป็นจุดที่ใช้ติดต่อกับสตรีมเพื่อรับหรือส่งข้อมูลก็ได้

สตรีมแบบตามลักษณะข้อมูลจะแบ่งออกเป็น 2 ประเภท คือ

1. สตรีมเท็กซ์ (text stream) หรือ เท็กซ์ไฟล์ (text file)
2. สตรีมไบนารี (binary stream) หรือ ไบนารีไฟล์ (binary file)

1. สตรีมเท็กซ์ (text stream) เป็นไฟล์ที่จัดเก็บข้อความ ซึ่งมีคุณลักษณะที่สำคัญคือ จะบันทึกข้อมูลที่เป็นข้อความต่าง ๆ ตามรหัสแอสกีของแต่ละตัวอักษร ดังนั้น สตรีมเท็กซ์ หรือเท็กซ์ไฟล์จึงสามารถถูกเปิดอ่านด้วยโปรแกรม Notepad และสามารถอ่านข้อความที่บันทึกไว้อย่างเข้าใจ อย่างไรก็ตาม สำหรับเรคคอร์ดที่ถูกบันทึกลงในเท็กซ์ไฟล์นั้น ในแต่ละเรคคอร์ดจะต้องถูกปิดท้ายด้วยรหัสปิดขึ้นบรรทัดใหม่ (\n) ด้วยเพื่อให้รับรู้ว่าเป็นข้อมูลของเรคคอร์ดนั้น ๆ

2. **สตรีมไบนารี(binary stream)** เป็นแฟ้มข้อมูลที่จัดเก็บข้อมูลชนิดเลขฐานสอง ดังนั้น สตรีมไบนารีหรือไบนารีไฟล์ เมื่อถูกเปิดด้วยโปรแกรม notepad แล้ว จะเป็นรหัสข้อมูลต่าง ๆ ที่เราไม่สามารถอ่านได้ เนื่องจากเป็นภาษาเครื่องหรือเลขฐาน 2 นั่นเอง

และสิ่งสำคัญของไฟล์ทั้ง 2 ลักษณะที่เก็บข้อมูลทั้ง 2 แบบนั้น ข้อมูลแต่ละไบต์จะถูกเก็บเรียงกันไปและจบท้ายด้วยรหัส EOF (End of File)



การทำงานกับเท็กซ์ไฟล์

1. ตัวแปรพอยน์เตอร์

ในการเขียนโปรแกรมเพื่อสร้างหรืออ่านไฟล์กับแผ่นดิสก์หรือหน่วยความจำสำรอง จะต้องใช้ตัวแปรตัวหนึ่งที่มีชื่อว่า ไฟล์พอยน์เตอร์ (File Pointer) ตัวแปรตัวนี้จะบอกว่าตำแหน่งที่เรา กำลังจะดำเนินการอยู่นี้อยู่ที่ตำแหน่งใดของไฟล์ เช่น ต้นไฟล์ กลางไฟล์ หรือท้ายไฟล์ ถ้าหากไม่มีตัวแปรไฟล์พอยน์เตอร์เราจะไม่สามารถกระทำการใด ๆ กับไฟล์ได้

รูปแบบคำสั่ง การประกาศเพื่อกำหนดตัวแปรทำหน้าที่เป็นไฟล์พอยน์เตอร์

```
File *ชื่อตัวแปรไฟล์พอยน์เตอร์
```

2. การเปิดไฟล์

การเปิดไฟล์บนแผ่นดิสก์หรือหน่วยความจำสำรอง จะใช้ฟังก์ชัน fopen() อยู่ในเฮดเดอร์ไฟล์ที่ชื่อ stdio.h โดยมีรูปแบบการใช้งานคำสั่ง ดังนี้

```
ตัวแปรไฟล์ = fopen(“ชื่อไฟล์”, “รูปแบบการเปิดไฟล์”);
```

โดยจะเปิดไฟล์ตามที่กำหนดไว้ในชื่อไฟล์ซึ่งอาจเป็นชื่อไฟล์คงที่หรือเป็นตัวแปรสตริงก็ได้ โดยสามารถระบุใคร่ที่เก็บข้อมูลลงไปด้วยก็ได้ ส่วนรูปแบบการเปิดไฟล์จะเป็นตัวบอกว่าไฟล์ที่เปิดนั้นเราต้องการทำอะไร โดยมีรูปแบบดังตารางต่อไปนี้

โหมด	การทำงาน
r	เปิดเท็กซ์ไฟล์เพื่ออ่าน
w	สร้างเท็กซ์ไฟล์ใหม่เพื่อเขียน
a	เปิดเท็กซ์ไฟล์เพื่อเขียนข้อมูลต่อท้าย
rb	เปิดไบนารีไฟล์เพื่ออ่าน
wb	สร้างไบนารีเพื่อเขียน
ab	สร้างไบนารีเพื่อเขียนข้อมูลต่อท้าย
r+	เปิดเท็กซ์ไฟล์เพื่ออ่านหรือเขียนทับไฟล์เก่า
w+	เปิดเท็กซ์ไฟล์เพื่ออ่านหรือเขียนทับไฟล์เก่า
a+	เปิดเท็กซ์ไฟล์เพื่อต่อท้ายไฟล์เก่า หรือเขียนไฟล์ใหม่
r+b	เปิดไบนารีไฟล์เพื่ออ่านหรือเขียนทับไฟล์เก่า
w+b	เปิดไบนารีไฟล์เพื่ออ่านหรือเขียนทับไฟล์เก่า หรือไฟล์ใหม่
a+b	เปิดไบนารีไฟล์เพื่อเขียนต่อท้ายไฟล์เก่า หรือเขียนไฟล์ใหม่

ตัวอย่างที่ 1

```
ft=fopen("data.txt", "w")
```

หมายถึง เปิดไฟล์ชื่อ data.txt เพื่อการเขียน ถ้าหากในหน่วยความจำสำรองไม่มีไฟล์ชื่อนี้ ก็จะเป็นการสร้างไฟล์นี้ขึ้นมาใหม่ แต่ถ้ามีไฟล์อยู่แล้วก็จะถูกเขียนทับ

ตัวอย่างที่ 2

```
ft=fopen("d:\data.txt", "w")
```

หมายถึง เปิดไฟล์ชื่อ data.txt เพื่อการเขียนที่อยู่ไดรฟ์ D ถ้าหากในหน่วยความจำสำรองไม่มีไฟล์ชื่อนี้ ก็จะเป็นการสร้างไฟล์นี้ขึ้นมาใหม่ แต่ถ้ามีไฟล์อยู่แล้วก็จะถูกเขียนทับ

ตัวอย่างที่ 3

```
ft=fopen("data.txt", "a+")
```

หมายถึง เปิดไฟล์ชื่อ data.txt เพื่อการเขียนข้อมูลเพิ่มท้ายไฟล์เก่า ถ้าหากไม่มีไฟล์อยู่ ก็จะเป็นการสร้างไฟล์ขึ้นมาใหม่

ตัวอย่างที่ 4

```
ft=fopen("data.txt", "r")
```

หมายถึง เปิดไฟล์ชื่อ data.txt เพื่ออ่านข้อมูล ถ้าหากไม่มีไฟล์อยู่จะเกิดข้อผิดพลาดขึ้น

ถ้าไฟล์นั้นเปิดได้จะมีการส่งค่าชี้ข้อมูลไฟล์ (file pointer) แต่ถ้าไฟล์นั้นไม่สามารถเปิดได้จะมีการส่งค่า NULL กลับออกมา ดังนั้นเมื่อมีการเปิดไฟล์ด้วยคำสั่ง fopen() จะต้องมีการตรวจสอบก่อนที่จะดำเนินการต่อไปว่าไฟล์นั้นเปิดได้หรือไม่ ดังตัวอย่างโค้ดดังต่อไปนี้

ในการเปิดไฟล์หากตัวแปร ft มีค่าเท่ากับ NULL แสดงว่าการเปิดไฟล์ไม่สำเร็จ ดังนั้นหากมีกรณีที่เกิด Error ขึ้นเกี่ยวกับการเปิดไฟล์หรือการทำงานแล้วต้องการให้โปรแกรมออกจากการทำงานให้ใช้คำสั่ง exit() ได้ดังนี้

- 1) หากต้องการโปรแกรมจบการทำงานในกรณีที่โปรแกรมทำงานสมบูรณ์ แล้วต้องการออกจากโปรแกรมให้ใช้คำสั่ง exit(0) เพื่อทำการออกจากโปรแกรม
- 2) หากต้องการโปรแกรมจบการทำงานในกรณีที่โปรแกรมทำงานไม่สมบูรณ์ แล้วต้องการออกจากโปรแกรมให้ใช้คำสั่ง exit(1) เพื่อทำการออกจากโปรแกรม

3. การปิดไฟล์

หลังจากไฟล์นั้นเปิดได้และดำเนินการต่าง ๆ กับไฟล์ไปแล้ว จะต้องปิดไฟล์ เพื่อป้องกันข้อมูลสูญหาย โดยใช้ฟังก์ชัน fclose() โดยมีรูปแบบดังนี้

รูปแบบที่ 1

```
int fclose(FILE *ชื่อตัวแปรไฟล์พอยน์เตอร์)
```

รูปแบบที่ 2

`fclose(ชื่อตัวแปรไฟล์พอยน์เตอร์)`

เมื่อระบบสามารถปิดไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ชื่อจะคืนค่า 0 คืนออกมานั่นเอง

ตัวอย่างที่ 1

```
int fclose(FILE *ft)
```

หรือ `fclose(ft)`

หมายถึง ปิดไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ชื่อ ft ซึ่งตำแหน่งอยู่



กิจกรรมเสริมทักษะ

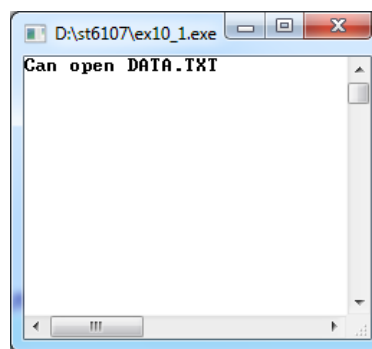
ให้นักเรียนทดลองเขียน โปรแกรมที่มีการแสดงการเปิดไฟล์ในโหมด “w” และให้บันทึก

ไฟล์ชื่อ ex10_1.cpp

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4 main()
5 {
6 FILE *fp;
7 fp=fopen("data.txt","w");
8 if (ft==NULL)
9 {
10 printf("Cannot open DATA.TXT\n");
11 exit(1);
12 }
13 else
14 {
15 printf("Can open DATA.TXT\n");
16 }
17 fclose(fp);
18 getch();
19 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรม ex10_1.exe



การใช้คำสั่งทำงานข้อมูลกับเท็กซ์ไฟล์

1. การใช้ฟังก์ชัน fprintf()

การใช้ฟังก์ชัน `fprintf()` เหมือนกับฟังก์ชัน `printf()` ตรงที่นำข้อมูลมาแสดงผล แต่ต่างกันที่การนำข้อมูลไปแสดงผลลงไฟล์ (บันทึก) หรือหน้าจอตัวเอง โดยมีรูปแบบการใช้งานฟังก์ชัน `fprintf()` ดังนี้

รูปแบบ

```
fprintf(ชื่อตัวแปรไฟล์พอยน์เตอร์, "รูปแบบชนิดข้อมูล", ชื่อตัวแปร);
```

ตัวอย่างที่ 1

```
fprintf(fp, "%d", age);
```

หมายถึง การนำค่าจากตัวแปรที่ชื่อ `age` ไปแสดงผลหรือบันทึกลงในไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ที่ชื่อ `fp` ซึ่งตำแหน่งอยู่

ตัวอย่างที่ 2

```
fprintf(fp, "%s %d %c", name, age, sex);
```

หมายถึง การนำค่าจากตัวแปรที่ชื่อ `name`, `age` และ `sex` ไปแสดงผลหรือบันทึกลงในไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ที่ชื่อ `fp` ซึ่งตำแหน่งอยู่

2. การใช้ฟังก์ชัน fscanf()

การใช้ฟังก์ชัน `fscanf()` เหมือนกับฟังก์ชัน `scanf()` ตรงที่เป็นการรับข้อมูล แต่ต่างกันที่การรับข้อมูลจากไฟล์หรือคีย์บอร์ดนั่นเอง โดยมีรูปแบบการใช้งานฟังก์ชัน `fscanf()` ดังนี้

รูปแบบ

```
fscanf(ชื่อตัวแปรไฟล์พอยน์เตอร์, "รูปแบบการแสดงผล", ชื่อตัวแปร);
```

ตัวอย่างที่ 1

```
fscanf(fp, "%d", &age);
```

หมายถึง การรับค่าจากไฟล์ที่ไฟล์พอยน์เตอร์ชื่อ fp ซึ่งตำแหน่งอยู่มาเก็บไว้ในตัวแปร
ที่ชื่อ age จาก

ตัวอย่างที่ 2

```
fscanf(fp, "%d %c", &age, &sex);
```

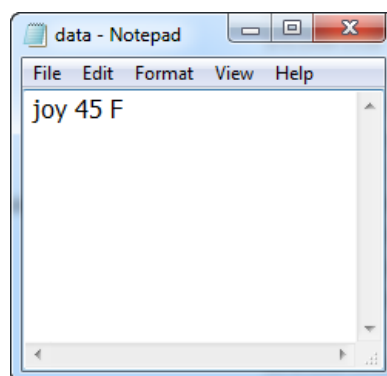
หมายถึง การรับค่าจากไฟล์ที่ไฟล์พอยน์เตอร์ชื่อ fp ซึ่งตำแหน่งอยู่ มาเก็บไว้ในตัวแปร
ที่ชื่อ name, age และ sex

กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียน โปรแกรมที่มีการแสดงการรับและแสดงผลจากไฟล์ข้อมูล และ
ให้บันทึกไฟล์ชื่อ ex10_2.cpp

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4 main()
5 {
6 char a[20]="joy",c='F',
7 char name[20],sex;
8 int b=45;
9 int age;
10 FILE *fp;
11 fp=fopen("d:\\data.txt","w");
12 fprintf(fp,"%s\n %d\n %c\n",a,b,c);
13 fclose(fp);
14 fp=fopen("d:\\data.txt","r");
15 fscanf(fp,"%s %d %c",name,&age,&sex);
16 printf("\n %s %d %c",name,age,sex);
17 fclose(fp);
18 getch();
19 }
```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรม ex10_2.exe และ notepad

3. การใช้ฟังก์ชัน fputc()

เป็นฟังก์ชันที่ทำงานคล้ายกับฟังก์ชัน putchar() เป็นการนำตัวอักษรจำนวน 1 ไบต์ไปบันทึกไว้ในไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ชี้อยู่ โดยมีรูปแบบการใช้งานดังนี้

รูปแบบ

```
fputc( 'อักขระ', ชื่อไฟล์พอยน์เตอร์);
```

ตัวอย่างที่ 1

```
fputc ( 'A' , fp);
```

หมายถึง การนำอักขระ 'A' ไปแสดงผลหรือบันทึกลงในไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ที่ชื่อ fp ชี้อยู่

ตัวอย่างที่ 2

```
fputc ( 'A' , FILE *fp);
```

หมายถึง การนำอักขระ 'A' ไปแสดงผลหรือบันทึกลงในไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ที่ชื่อ fp ชี้อยู่

4. การใช้ฟังก์ชัน fgetc()

เป็นฟังก์ชันที่ทำงานคล้ายกับ getchar() เป็นการรับอักขระจำนวน 1 ไบต์จากไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ชี้อยู่ โดยมีรูปแบบการใช้งานดังนี้

รูปแบบ

```
fgetc( ชื่อไฟล์พอยน์เตอร์);
```

ตัวอย่างที่ 1

```
ans=fgetc ( fp);
```

หมายถึง การนำอักขระจากไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ชี้อยู่มาเก็บไว้ในตัวแปร ans

ตัวอย่างที่ 2

```
printf("%c",fgetc(fp));
```

หมายถึง แสดงค่าของอักขระที่บันทึกอยู่ในไฟล์ที่ตัวแปรไฟล์พอยน์เตอร์ที่ชื่อ fp ชี้อยู่



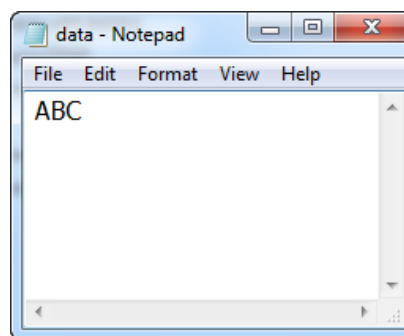
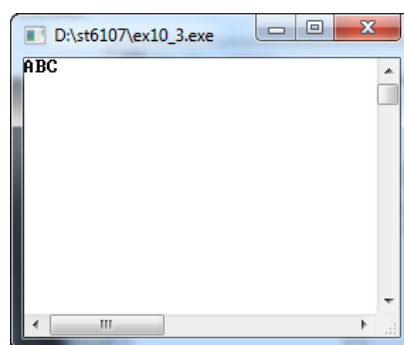
กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียน โปรแกรมที่มีการแสดงการรับและแสดงผลอักขระจากไฟล์ข้อมูล และให้บันทึกไฟล์ชื่อ ex10_3.cpp

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4 char ans;
5 main()
6 {
7 FILE *fp;
8 fp=fopen("d:\\data.txt","w");
9 if(fp==NULL)
10 {
11 printf("Can not open file\n");
12 exit(0);
13 }
14 fputc('A' , fp);
15 fputc('B' , fp);
16 fputc('C' , fp);
17 fclose(fp);
18 fp=fopen("d:\\data.txt","r");
19 ans=fgetc(fp); printf("%c",ans);
20 ans=fgetc(fp); printf("%c",ans);
21 ans=fgetc(fp); printf("%c",ans);
22 fclose(fp);
23 getch();
24 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex10_3.exe และ notepad

4. การใช้ฟังก์ชัน fputs()

เป็นฟังก์ชันที่ทำงานคล้ายกับ fprintf() คือ สามารถเขียนข้อมูลประเภทข้อความ (String) ลงไปในไฟล์ได้ แต่ fprintf() มีจุดเด่นคือสามารถกำหนดข้อมูลลงไปในตัวแปรได้ด้วย

รูปแบบ

```
fputs( "ข้อความ" , ชื่อตัวแปรไฟล์พอยน์เตอร์);
```

ตัวอย่าง

```
FILE *fp;
fp=fopen("d:\\data.txt","w");
fputs("Chaibadanwithhaya\n",fp);
```

หมายถึง การนำข้อความคำว่า “Chaibadanwithhaya” ที่จำนวนตัวอักษร 17 ตัวเก็บไว้ในไฟล์ที่ชื่อว่า data.txt โดยมีตัวแปรไฟล์พอยน์เตอร์ที่ชื่อว่า fp ซึ่งตำแหน่งอยู่

4. การใช้ฟังก์ชัน fgets()

เป็นฟังก์ชันที่ทำงานคล้ายกับ fscanf() คือ สามารถรับข้อมูลประเภทข้อความ (String) ได้ แต่คำสั่ง fgets() จะเป็นการรับข้อความจากไฟล์ได้ที่ตัวแปรไฟล์พอยน์เตอร์ที่อยู่โดยอาศัยโครงสร้างการจับเก็บข้อมูลในรูปแบบของอาร์เรย์เวลารับค่ามาต้องรับค่าในรูปแบบอาร์เรย์เช่นกัน

รูปแบบ

```
fgets( "ข้อความ" , จำนวนตัวอักษรที่ต้องการ ,ชื่อตัวแปรอาร์เรย์);
```

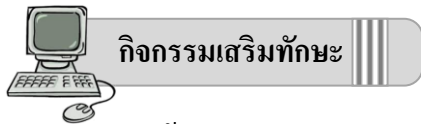
ตัวอย่าง

```
fgets(ans , 18 , fp);
printf("%s",ans);
```

หมายถึง การนำข้อความ ที่จำนวนตัวอักษร 17 ตัวที่เก็บไว้ในไฟล์ที่ชื่อว่า data.txt โดยมีตัวแปรไฟล์พอยน์เตอร์ที่ชื่อว่า fp ซึ่งตำแหน่งอยู่มาเก็บไว้ในตัวแปรที่ชื่อ ans แล้วแสดงค่าที่เก็บไว้ในตัวแปร ans ให้แสดงผลลัพธ์ได้

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ans[n]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ข้อมูล	C	h	a	i	b	a	d	a	n	w	i	t	t	h	a	y	a	\n

การจำลองหน่วยความจำในการจัดเก็บข้อความคำว่า Chaibadanwithhaya ในไฟล์ที่ชื่อว่า data.txt

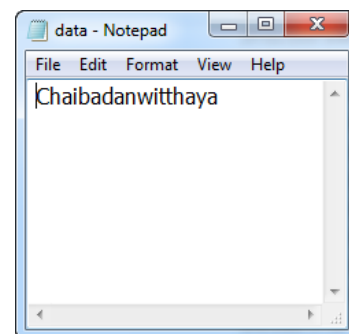
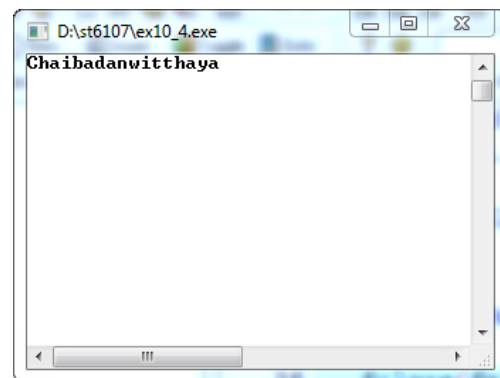


ให้นักเรียนทดลองเขียน โปรแกรมที่มีการแสดงการรับและแสดงผลอักขระจากไฟล์ข้อมูล และให้บันทึกไฟล์ชื่อ ex10_4.cpp

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4 char ans[18];
5 main(void)
6 {
7     FILE *fp;
8     fp=fopen("d:\data.txt","w");
9     if(fp==NULL)
10    {
11        printf("Can not open file\n");
12        exit(0);
13    }
14    fputs("Chaibadanwithhaya\n",fp);
15    fclose(fp);
16    fp=fopen("d:\data.txt","r");
17    while(!feof(fp))
18    {
19        fgets(ans,18,fp);
20        printf("%s",ans);
21    }
22    fclose(fp);
23    getch();
24 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex10_4.exe และ notepad