

ใบงานการทดลองที่ 15  
เรื่อง การจัดการไฟล์ข้อมูล

1. จุดประสงค์ทั่วไป

3.1. รู้และเข้าใจแนวทางการประยุกต์การเขียนโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

3.1.5. ออกแบบแนวทางการแก้ไขปัญหาการจัดการไฟล์ข้อมูล

3.1.6. วิเคราะห์แนวทางการจัดการไฟล์ข้อมูล

3.1.7. วางหลักการโครงสร้างการจัดการไฟล์ข้อมูล

3.1.8. ฝึกหัดและทดลองการจัดการไฟล์ข้อมูล

3.1.9. แก้ไขและประยุกต์ใช้งานการจัดการไฟล์ข้อมูลร่วมกับการจัดการฟังก์ชัน

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

4.1. จงบอกและอธิบายการใช้งานคำสั่งเปิดไฟล์ข้อมูล

มใช้จวน File สามารถเปิดได้จวน library stdlib โดยใส่  
คำสั่ง fopen ในมใช้จวน โดยใส่ file และลักษณะการอ่าน r กับ r+  
คือมอ่าน w กับ w+ คือมเขียนจวน a กับ a+ คือมเขียนจวน

4.2. จงบอกและอธิบายการใช้งานคำสั่งปิดไฟล์ข้อมูล

fclose ตามด้วย File pointer คำสั่งเพื่อปิดมอ่าน

4.3. จงบอกและอธิบายการใช้งานโหมดการทำงานเกี่ยวกับไฟล์ทุกชนิด

r ก็คืออ่านไฟล์  
r+ ก็คืออ่านและเขียนไฟล์  
w ก็คือเขียนไฟล์ใหม่  
w+ คือ เขียนและอ่านไฟล์ใหม่จากไฟล์ที่ว่าง  
a คือ เขียนจวนต่อจากเดิม  
a+ คือ เขียนและอ่านจวนต่อจากเดิม

4.4. จงบอกและอธิบายการใช้งานคำสั่งอ่านไฟล์ข้อมูล (พร้อมยกตัวอย่างประกอบ)

พยาน. คำสั่งอ่านไฟล์ ใช้ฟังก์ชัน `fscanf` ใน `stdio.h` เพื่ออ่านข้อมูลจากไฟล์ตามรูปแบบที่กำหนดไว้

เช่น `fscanf(f, "%s %s %s", str1, str2, str3);`  
จากตัวอย่างจะเห็นว่า `fscanf` จะอ่านข้อมูลจากไฟล์ตามรูปแบบที่กำหนดไว้

4.5. จงบอกและอธิบายการใช้งานคำสั่งเขียนไฟล์ข้อมูล (พร้อมยกตัวอย่างประกอบ)

พยาน. คำสั่งเขียนไฟล์ ใช้ฟังก์ชัน `fprintf` ใน `stdio.h` เพื่อเขียนข้อมูลลงในไฟล์ตามรูปแบบที่กำหนดไว้

เช่น `fopen` ใช้ในฟังก์ชัน `fprintf` เพื่อเปิดไฟล์ก่อนเขียนข้อมูล  
เช่น `fprintf(f, "%s %s %s", str1, str2, str3);`  
จากตัวอย่างจะเห็นว่า `fprintf` จะเขียนข้อมูลลงในไฟล์ตามรูปแบบที่กำหนดไว้

4.6. จงบอกและอธิบายการสร้างและใช้งานฟังก์ชันการอ่านไฟล์ข้อมูล

พยาน. คำสั่งสร้างและใช้งานฟังก์ชันการอ่านไฟล์ ใช้ฟังก์ชัน `fopen` ใน `stdio.h` เพื่อเปิดไฟล์ก่อนอ่านข้อมูล  
เช่น `fscanf` เพื่ออ่านข้อมูลจากไฟล์

4.7. จงบอกและอธิบายการสร้างและใช้งานฟังก์ชันการเขียนไฟล์ข้อมูล

พยาน. คำสั่งสร้างและใช้งานฟังก์ชันการเขียนไฟล์ ใช้ฟังก์ชัน `fopen` ใน `stdio.h` เพื่อเปิดไฟล์ก่อนเขียนข้อมูล  
เช่น `fprintf` เพื่อเขียนข้อมูลลงในไฟล์

## 5. ลำดับขั้นตอนการปฏิบัติงาน

5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.1.1. จากไฟล์ Salary.txt จงเขียนโปรแกรมเพื่อค้นหาผู้ที่มีรายได้มากที่สุดในสาขาอาชีพ

Salary.txt

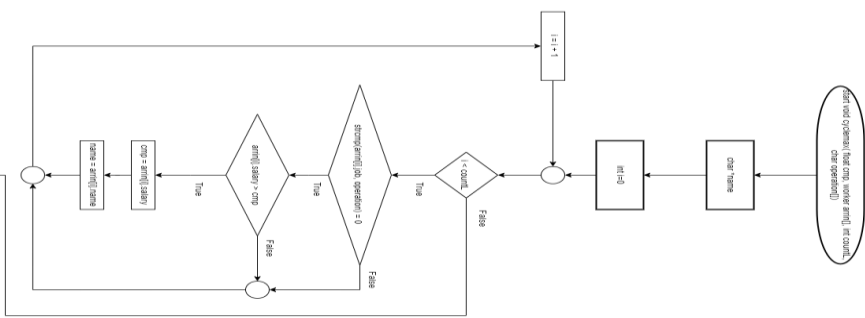
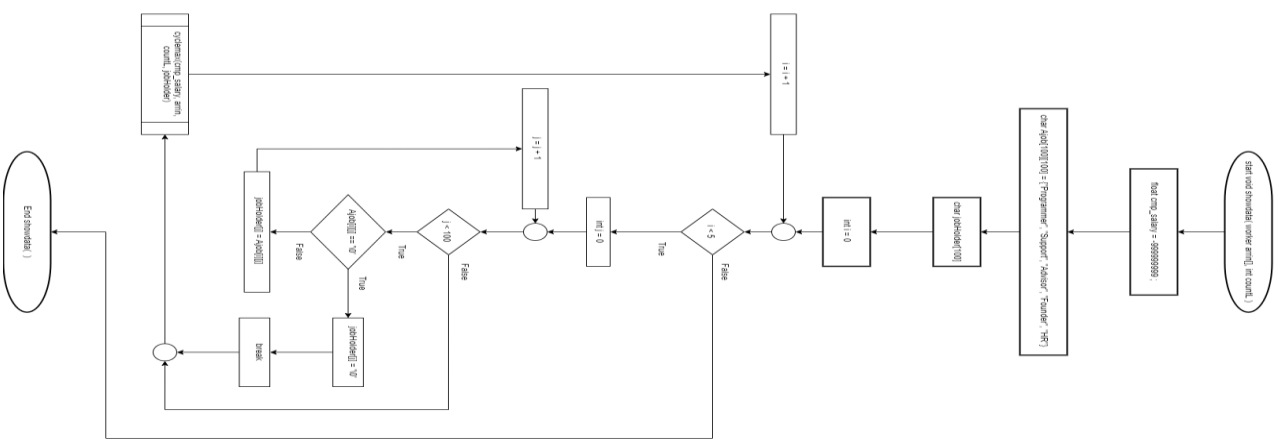
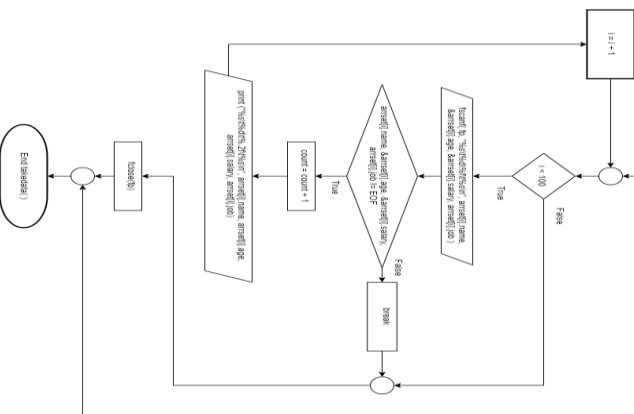
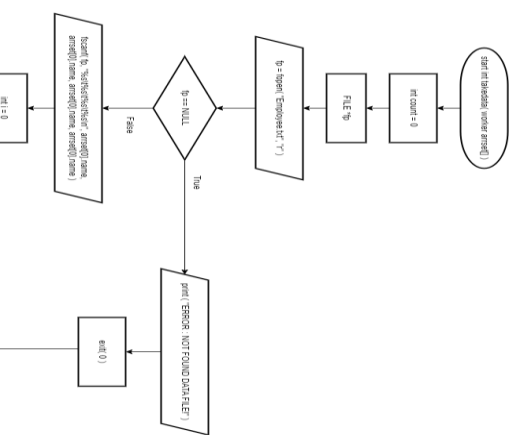
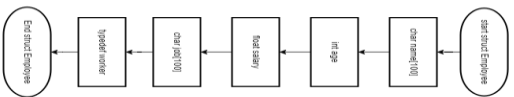
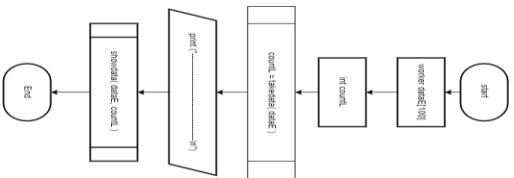
Name	Age	Salary(Bath)	Job
Kim	25	31,250	Advisor
Pong	32	45,000	Programmer
Som	25	23,000	Support
Aoy	24	33,250	Advisor
Ying	26	34,550	Programmer
DJ	29	21,1500	HR
Yot	28	50,000	Founder
Pot	31	24,500	HR
Vip	25	25,450	Programmer
A	21	22,250	Support

### Test case 1

Input	
Output	Programmer : Pong 45,000 bath Support : Som 23,000 bath Advisor : Aoy 33,250 bath Founder : Yot 50,000 bath HR : Pot 24,500 bath

### 5.1.2. จงเขียนผังงาน

ผังงาน



### 5.1.3. จงเขียนโค้ดโปรแกรม

#### โค้ดโปรแกรม

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4
5  struct employee {
6      char name[100] ;
7      int age ;
8      float salary ;
9      char job[100] ;
10 }typedef worker ;
11
12 int collector (worker [] ) ;
13 void sortAge( worker [], int ) ;
14 void sortSalary( worker [], int ) ;
15 void search( worker [], int, int ) ;
16
17 int main() {
18     FILE *fp ;
19     int countL ;
20     int searcher ;
21     worker employee[100] ;
22     countL = collector(employee) ;
23     printf("-----\n") ;
24     printf("Find Age : ") ;
25     scanf( "%d", &searcher ) ;
26     sortAge( employee, countL ) ;
27     sortSalary( employee, countL ) ;
28     search( employee, countL, searcher) ;
29 }
```

```

31 int collector (worker take[] ) {
32     int count = 0 ;
33     FILE *fp ;
34     fp = fopen( "Employee.txt", "r" );
35     if (fp == NULL ){
36         printf( "ERROR : NOT FOUND DATA FILE!" );
37         exit( 0 ) ;
38     }
39     fscanf( fp, "%s\t%s\t%s\t%s\n", take[0].name, take[0].name, take[0].name, take[0].name ) ;
40
41     for (int i = 0 ; i < 100 ; i++){
42
43         if (fscanf( fp, "%s\t%d\t%f\t%s\n", take[i].name, &take[i].age, &take[i].salary, take[i].job ) != E
44             count++ ;
45             printf("%s\t%d\t%.2f\t%s\n", take[i].name, take[i].age, take[i].salary, take[i].job) ;
46         }
47         else{
48             break ;
49         }
50     }
51     fclose(fp) ;
52     return count ;
53 }
54
55 void sortAge( worker data[], int num_worker ) {
56     worker holdTposi ;
57     for (int i = 1; i < num_worker ; i++) {
58         for (int j = 0 ; j < i ; j++) {
59             if( data[i].age > data[j].age){
60                 holdTposi = data[i] ;

```

Activate Windows  
Go to Settings to activate Windows

```

60                 holdTposi = data[i] ;
61                 for( int k = i ; k > j ; k--){
62                     data[k] = data[k-1] ;
63                 }
64                 data[j] = holdTposi ;
65                 break ;
66             }
67         }
68     }
69
70     printf("OUTPUT:\n\t\t\t" ) ;
71     for (int c = 0 ; c < 5 ; c++){
72         printf("%d\t", data[c].age) ;
73     }
74     printf("\n" ) ;
75     printf("Age( 99 - 0 )\t" ) ;
76     for (int l = 0 ; l < 5 ; l++){
77         printf("%s\t", data[l].name) ;
78     }
79     printf("\n" ) ;
80 }
81
82 void sortSalary( worker data[], int num_worker ) {
83     worker holdTposi ;
84     for (int i = 1; i < num_worker ; i++) {
85         for (int j = 0 ; j < i ; j++) {
86             if( data[i].salary < data[j].salary){
87                 holdTposi = data[i] ;
88                 for( int k = i ; k > j ; k--){

```



```

89         data[k] = data[k-1] ;
90     }
91     data[j] = holdTposi ;
92     break ;
93 }
94 }
95
96 }
97 printf("OUTPUT:\n\t\t\t\t") ;
98 ✓ for (int c = 0 ; c < 5 ; c++){
99     | printf("%.f\t", data[c].salary) ;
100 }
101 printf("\n") ;
102 printf("Salary( 99999 - 0 )\t") ;
103 ✓ for (int l = 0 ; l < 5 ; l++){
104     | printf("%s\t", data[l].name) ;
105 }
106 printf("\n") ;
107 }
108
109 ✓ void search( worker data[], int num_worker, int founder ) {
110     | printf( "\nAge %d = ", founder ) ;
111     | for ( int i = 0 ; i < num_worker ; i++) {
112     |     | if( data[i].age == founder ){
113     |     |     | printf( "%s\t", data[i].name ) ;
114     |     |     }
115     |     }
116 }

```

## 6. สรุปผลการปฏิบัติงาน

จากข้อมูล รส. data.y.txt เวลาอ่านได้ค่าเฉลี่ยของมวลกระดูก  
แล้วจะเห็นว่า ได้ 91.96 เปอร์เซ็นต์

## 7. คำถามทางการทดลอง

7.1. จงระบุข้อควรระวังในการอ่านไฟล์ข้อมูล

ก่อนจะรันโปรแกรม ต้องดูว่าไฟล์ที่เราจะอ่านมีชื่อ file  
หรือไม่ ในที่นี้ดูว่าไฟล์ที่เราจะอ่านมีชื่อหรือไม่

7.2. จงระบุข้อควรระวังในการเขียนไฟล์ข้อมูล

ก่อนจะ place holder ต้องดูว่าชื่อของข้อมูลที่เราจะใส่  
มีอยู่แล้วหรือไม่ ถ้ามีแล้วต้องใส่ชื่อใหม่

7.3. หากเลือกใช้งานคำสั่งอ่านไฟล์ข้อมูล แต่ไม่มีไฟล์ข้อมูลดังกล่าวอยู่ในระบบ ควรดำเนินการ  
อย่างไร ?

ถ้าไฟล์ที่เราจะอ่านไม่มีอยู่ในระบบ เราจะต้องดูว่า folder  
ที่มี file ของเราอยู่หรือไม่ ถ้าไม่มี เราจะต้องสร้าง folder  
ขึ้นมาใหม่ก่อนที่เราจะรันโปรแกรม

7.4. จงอธิบายแนวทางในการอ่านไฟล์ข้อมูลเพื่อบันทึกข้อมูลลงในตัวแปรโครงสร้างข้อมูล

ก่อนจะรันโปรแกรม เราจะต้องดูว่าชื่อของข้อมูลที่เราจะใส่  
มีอยู่แล้วหรือไม่ ถ้ามีแล้วต้องใส่ชื่อใหม่