



รายงาน : Lab05 - File

จัดทำโดย

นายกิตติพิศ หนูทอง รหัสนักศึกษา : 6135512003

นายปฏิภาณ วรรณโก รหัสนักศึกษา : 6135512059

Section : 01

240-207 Programing and Data Structures

“งานทั้งหมดนี้ในรายงานฉบับนี้ล้วนเป็นผลงานของข้าพเจ้า มิได้ลอกหรือสำเนาจากที่อื่นใด
ในกรณีที่พบว่าเกิดสำเนาด้วยวิธีใดก็ตาม ข้าพเจ้ายินดีไม่ขอรับคะแนนจากรายงานฉบับนี้”

คะแนนที่ได้

ลงชื่อ

กิตติพิศ หนูทอง

(นายกิตติพิศ หนูทอง)

ปฏิภาณ วรรณโก

(นายปฏิภาณ วรรณโก)

Lab05 - File

ข้อที่ 1 : Simple Text File

Simple Text File

ไฟล์ num.txt ในแต่ละบรรทัดจะมีตัวเลขจำนวนเต็ม 1 ตัวเลข จงเขียนโปรแกรมเพื่อหาผลรวมของตัวเลข
ตัวอย่างไฟล์ num.txt

```
7
2
1
```

รูปแบบการแสดงผล

```
=10
```

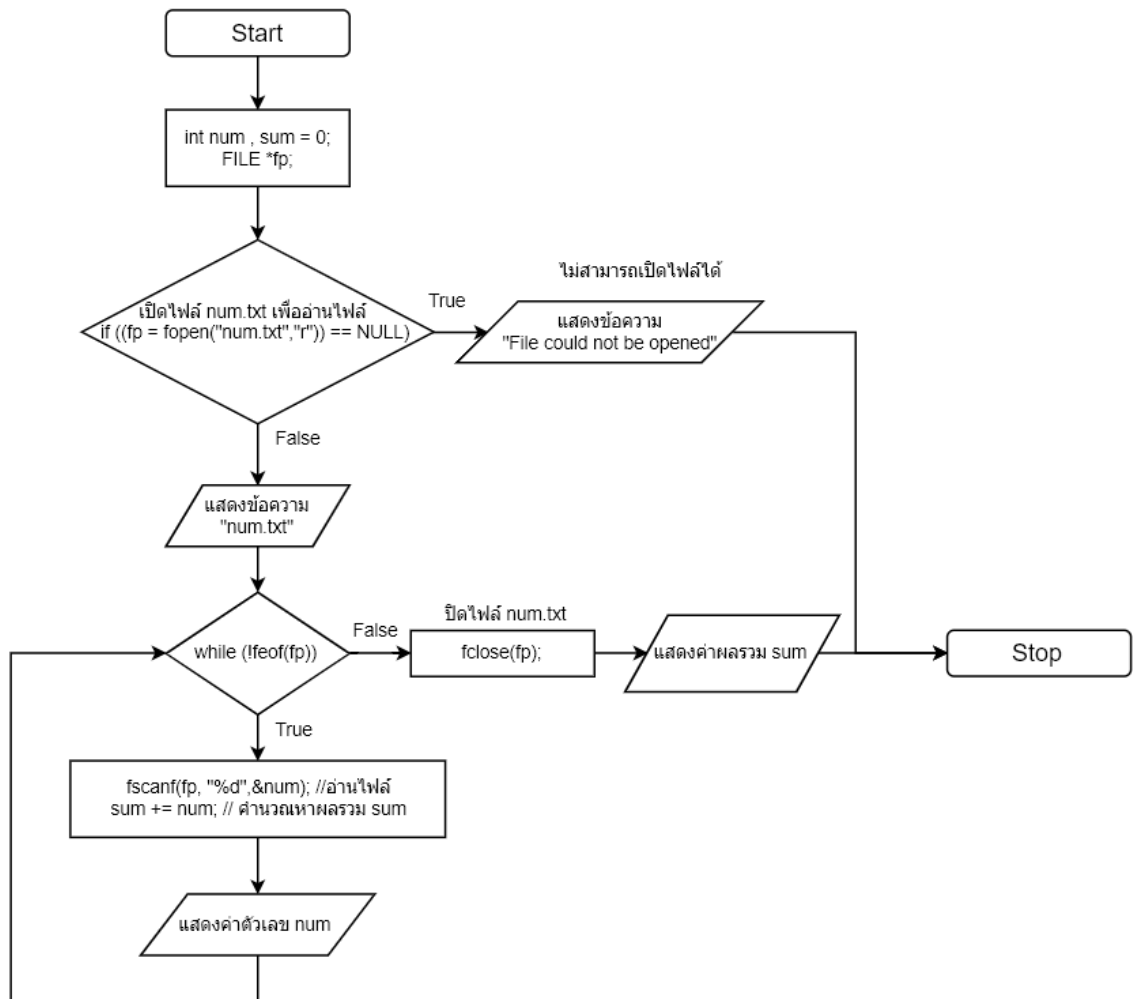
หากมีการแก้ไขไฟล์ num.txt อาจทำให้ `testing failed`.

Code

lab05-1.c

```
1  #include<stdio.h>
2  int main()
3  {
4      int num,sum=0;
5      FILE *fp;
6      if ((fp = fopen("num.txt","r")) == NULL)
7      {
8          printf("File could not be opened\n");
9      }
10     else
11     {
12         printf("num.txt\n");
13         while (!feof(fp))
14         {
15             fscanf(fp, "%d",&num);
16             sum += num;
17             printf("-> %d\n",num);
18         }
19         fclose(fp);
20         printf("sum = %d",sum);
21     }
22     return 0;
23 }
24
25 }
```

Flowchart



ข้อมูล ไฟล์ num.txt



num - Notepad

File Edit Format View Help

```
1
2
3
4
5
6
7
8
9
10|
```

ผลการรันโปรแกรม

```
C:\Users\User\OneDrive\Programming & Data\lab05\lab05-1.exe
num.txt
-> 1
-> 2
-> 3
-> 4
-> 5
-> 6
-> 7
-> 8
-> 9
-> 10
sum = 55
-----
Process exited after 0.1106 seconds with return value 0
Press any key to continue . . .
```

อธิบายหลักการทำงาน : โปรแกรมนี้จะเป็นการรับค่าจำนวนเต็มมาจากไฟล์ num.txt ซึ่งเป็นไฟล์ข้อความ และโปรแกรมจะทำการคำนวณหาผลรวมเลขจำนวนเต็มทั้งหมดจากไฟล์

หลักการทำงานจะมีการประกาศตัวแปร int จำนวนสองตัวได้แก่ num เพื่อรับค่าจำนวนเต็มมาจากไฟล์ และ sum เพื่อคำนวณค่าของผลรวม และมีประกาศ FILE pointer fp เพื่อให้มาเป็นค่าพอยน์เตอร์ที่ชี้ไปที่ไฟล์สำหรับการเปิดไฟล์หรืออื่นๆ เริ่มการทำงานของโปรแกรม โดยการให้ fp ชี้ไปที่ไฟล์ num.txt โดยใช้คำสั่งเปิดไฟล์ fopen เป็นโหมด “r” (โหมดอ่าน) และตรวจสอบว่าถ้าเปิดไฟล์แล้วเท่ากับ NULL ให้แสดงข้อความ “File could not be opened” (ไม่สามารถเปิดไฟล์ได้)

ถ้าไม่ใช่ตามเงื่อนไขดังกล่าวให้ทำการวนลูป while (!feof(fp)) โดยมีเงื่อนไขว่า วนลูปจนกว่าจะจบไฟล์ ซึ่งในลูปจะมีการใช้คำสั่ง fscanf(fp, "%d",&num); เพื่อเป็นการอ่านค่าจำนวนเต็มในไฟล์ (num.txt) และเก็บค่าไว้ในตัวแปร num ต่อจากนั้นจะใช้คำสั่ง sum += num; เพื่อคำนวณหาผลรวมของตัวเลขทุกจำนวน โดยที่ค่า num จะเปลี่ยนไปเรื่อยๆจากการอ่านไฟล์ ถัดไปจะแสดงค่าของ num ออกมาทีละตัววนลูปแบบนี้จนครบเงื่อนไข เมื่อออกจากลูปจะใช้คำสั่งปิดไฟล์ fclose(fp); และแสดงค่าของ sum ที่ทำการคำนวณผลรวมออกมาทางหน้าจอ

ความรู้จากการทำ Lab Simple Text File : ไฟล์ข้อความ (text) เป็นไฟล์ที่สามารถเปิดอ่านได้รู้เรื่องและเป็นไฟล์มาตรฐานในการบันทึกข้อความ ซึ่งสามารถเขียน อ่าน และแก้ไขต่อท้ายข้อความได้ โดยในการเขียนโปรแกรม การบันทึกข้อมูลลงในไฟล์ text จะใช้คำสั่ง fprintf และการอ่านข้อมูลจากไฟล์ text จะใช้คำสั่ง fscanf ซึ่งอาจมีข้อเสียในการอ่านค่าข้อมูลที่ไม่ใช่ชนิดที่เราต้องการ

ข้อที่ 2 : Array of Employee

Array of Employee

จงเขียนโปรแกรมที่ทำการอ่านไฟล์ไบนารี ชื่อ `employee.bin`

โดยไฟล์จะเริ่มต้นจากตัวเลขจำนวนเต็ม บอกจำนวนข้อมูลพนักงานที่บันทึกไว้ในไฟล์ จากนั้นประมวลผลข้อมูลให้อยู่ในรูปของ `struct employee` ที่กำหนดให้ แล้วทำการแสดงข้อมูลของพนักงานทุกคน พร้อมแสดงผลเงินเดือนรวม

โดยการอ่าน จะเป็นการอ่านเข้าสู่อาเรย์อย่างเหมาะสม - ข้อนี้มีการใช้งาน `malloc`

จะต้องมีการกด L-test อย่างน้อยหนึ่งครั้ง ก่อนการใช้ L-Execute เพื่อดาวน์โหลดไฟล์ที่เกี่ยวข้องโดยอัตโนมัติ

รูปแบบการแสดงผล

```
John Doe:3000.0
Mark Ken:2300.0
Sucy Merc:2000.0
=7300.0
```

Code

lab05-2.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct employee
4  {
5      char name[128];
6      float salary;
7  };
8  typedef struct employee Employee;
9
10 float total_salary(Employee all[], int len);
11
12 int main()
13 {
14     int num,i; //number of employees
15     Employee *all;
16     FILE *fp;
17
18     printf("Number of employee : ");
19     scanf("%d",&num);
20     all = (Employee *)malloc(num*sizeof(Employee));
```

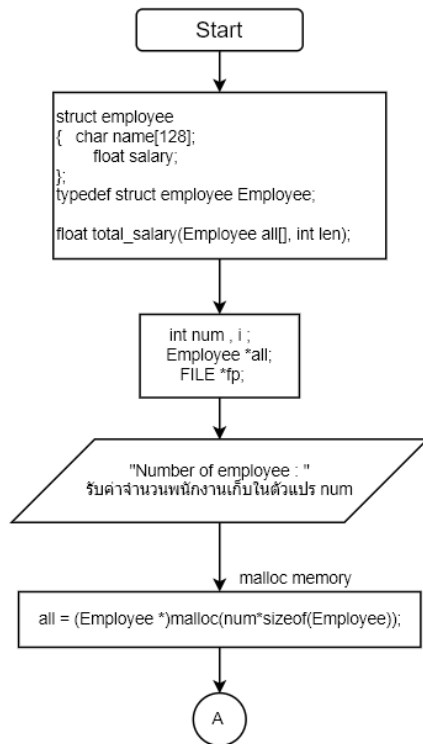
การบันทึกข้อความ ไฟล์ employee.bin

```
21 // Writing to binary file
22 if ((fp = fopen("employee.bin","wb")) == NULL)
23 {
24     printf("File could not be opened\n");
25 }
26 else
27 {
28     for (i = 0; i < num; i++)
29     {
30         fflush(stdin);
31         printf("Name : ");
32         scanf("%[^\n]s",all->name);
33         printf("Salary : ");
34         scanf("%f",&all->salary);
35         fwrite(all, sizeof(Employee),1,fp);
36     }
37     fclose(fp);
38 }
39
```

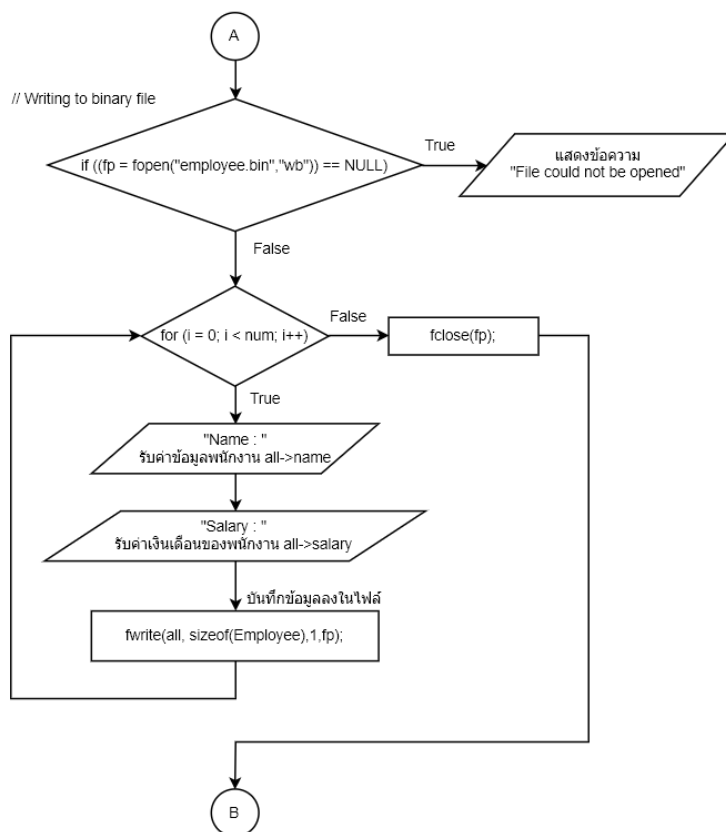
การอ่านข้อความ ไฟล์ employee.bin

```
40 // Reading from binary file
41 if ((fp = fopen("employee.bin","rb")) == NULL)
42 {
43     printf("File could not be opened\n");
44 }
45 else
46 {
47     fread(all, num*sizeof(Employee),1,fp);
48     for (i = 0; i < num; i++)
49     {
50         printf("%s : %.1f \n", (all+i)->name, (all+i)->salary);
51     }
52     fclose(fp);
53 }
54 printf("= %.1f",total_salary(all,num));
55 free(all);
56 return 0;
57 }
58
59 float total_salary(Employee all[], int len)
60 {
61     int i;
62     float sum=0;
63     for (i=0;i<len;i++)
64     {
65         sum += all[i].salary;
66     }
67     return sum;
68 }
```

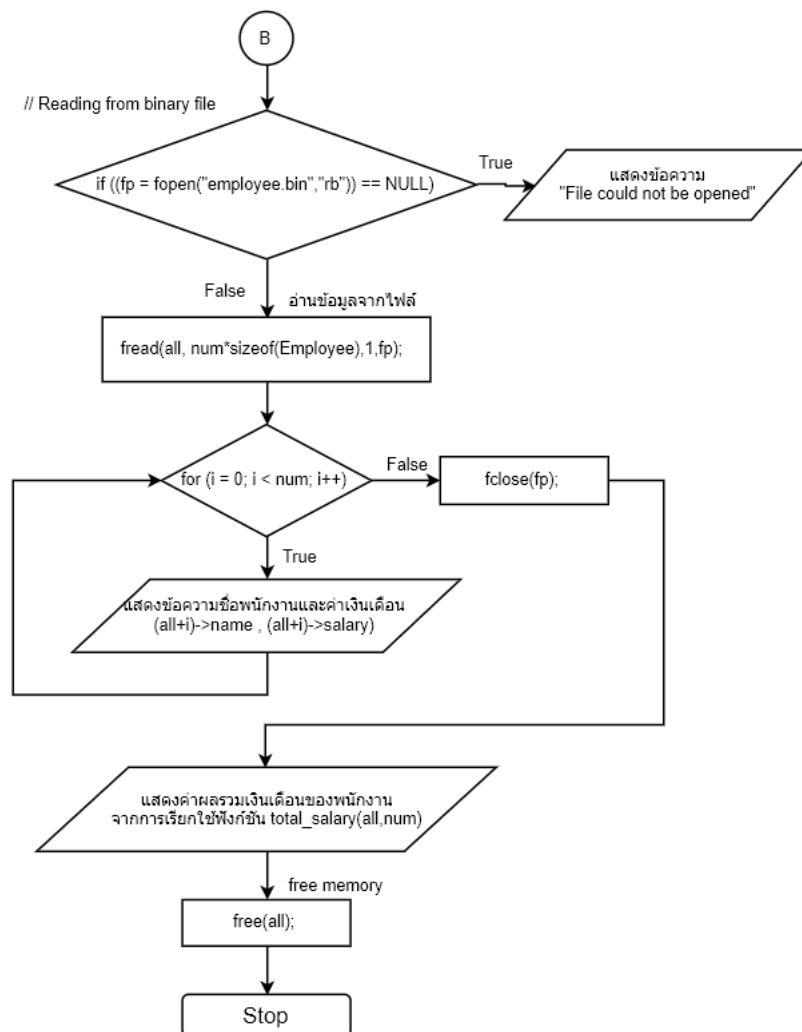
Flowchart



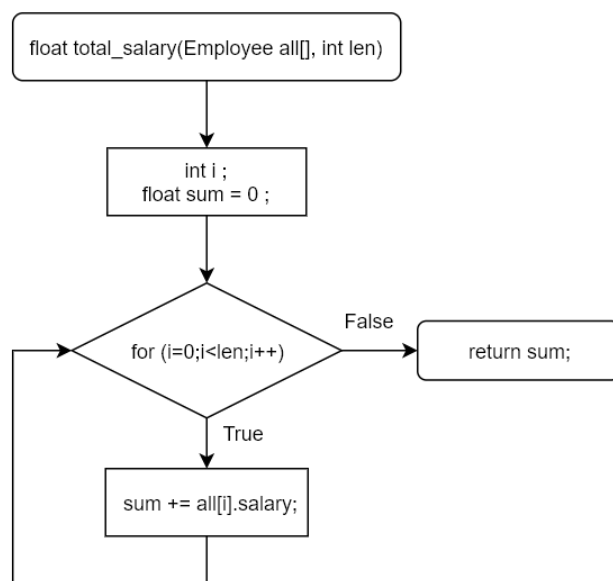
การบันทึกข้อความ ไฟล์ employee.bin



การอ่านข้อความ ไฟล์ employee.bin



ฟังก์ชัน float total_salary(Employee all[], int len)



ผลการรันโปรแกรม

- ผลรันแสดงการบันทึกข้อมูลพนักงานในไฟล์ employee.bin

```
C:\Users\User\OneDrive\Programming & Data\lab05\lab05-2.exe
Number of employee : 3
Name : Mark Ken
Salary : 15000
Name : Jim Barton
Salary : 25000
Name : Helen Ross
Salary : 20000
-----
Process exited after 159.1 seconds with return value 0
Press any key to continue . . .
```

- ผลรันแสดงการอ่านข้อมูลพนักงานจากไฟล์ employee.bin และแสดงผลการคำนวณผลรวมเงินเดือนของพนักงาน

```
C:\Users\User\OneDrive\Programming & Data\lab05\lab05-2.exe
Number of employee : 3
Mark Ken : 15000.0
Jim Barton : 25000.0
Helen Ross : 20000.0
= 60000.0
-----
Process exited after 0.8493 seconds with return value 0
Press any key to continue . . .
```

อธิบายหลักการทำงาน : เป็นโปรแกรมที่ทำการอ่านไฟล์ binary ชื่อ employee.bin ซึ่งในไฟล์จะมีโครงสร้างข้อมูล structure Employee ที่เก็บชื่อพนักงานไว้ในตัวแปร char name และเงินเดือนของพนักงานไว้ในตัวแปร float salary โดยที่โปรแกรมจะมีฟังก์ชันการทำงานสองฟังก์ชันได้แก่ ฟังก์ชัน main () และฟังก์ชัน float total_salary (Employee all[], int len) ซึ่งเป็นฟังก์ชันคำนวณหาผลรวมเงินเดือนของพนักงานที่ได้ค่าจากฟังก์ชัน main () ในโปรแกรมจะมีหลักการทำงานดังนี้

ฟังก์ชัน main () จะมีการประกาศตัวแปรประเภท int สองตัวได้แก่ num เป็นค่าที่จะรับจากผู้ใช้เพื่อกำหนดจำนวนพนักงานที่ต้องการจะทำการอ่านจากไฟล์ และตัวแปรประเภท Employee ที่เก็บข้อมูลพนักงานเป็นพอยน์เตอร์ชื่อ all ทำหน้าที่ในการเก็บข้อมูลที่ได้อ่านจากไฟล์ โดยที่โปรแกรมจะเริ่มจากแสดงข้อความ “num of employee” และรับค่าจำนวนพนักงานเก็บในตัวแปร num และทำการขอ memory ให้กับ

ตัวแปร all ในคำสั่ง `all = (Employee *) malloc (num*sizeof(Employee))` เพื่อสามารถรับข้อมูลที่กำหนดให้ได้ ซึ่งในโปรแกรมข้างต้นจะแบ่งเป็นการเขียนขึ้นสองแบบคือ แบบเขียนข้อมูลลงในไฟล์และการอ่านข้อมูลจากไฟล์

การเขียนข้อมูลบันทึกลงในไฟล์ : เริ่มจากการให้ fp ชี้ไปที่ไฟล์ employee.bin โดยใช้คำสั่ง `fopen` เป็นโหมด “wb” (โหมดเขียน) และเช็คการเปิดไฟล์ด้วยเงื่อนไข `if` ถ้าเปิดไฟล์และเท่ากับ NULL ให้แสดงข้อความ “File could not be opened” (ไม่สามารถเปิดไฟล์ได้) ถ้าไม่ตรงตามเงื่อนไขดังกล่าวให้ทำการวนลูปเงื่อนไข `for (i = 0; i < num; i++)` แล้วรับค่าที่ป้อนเข้ามา ชื่อพนักงานเก็บในตัวแปร `all->name` และเงินเดือนเก็บในตัวแปร `all->salary` หลังจากนั้นใช้คำสั่ง `fwrite(all, sizeof(Employee), 1, fp)`; คำสั่งนี้จะเป็นการบันทึกข้อมูลลงไปในไฟล์ตามขนาดที่กำหนดและตามจำนวนที่กำหนดลงในไฟล์ที่ fp ชี้ จากนั้นทำการวนลูปจนครบเงื่อนไขแล้วใช้คำสั่งปิดไฟล์ `fclose(fp)`; เป็นอันเสร็จในการเขียนข้อมูลลงในไฟล์

การอ่านข้อมูลจากไฟล์ : เริ่มจากการให้ fp ชี้ไปที่ไฟล์ employee.bin โดยใช้คำสั่ง `fopen` เป็นโหมด “rb” (โหมดอ่าน) และเช็คการเปิดไฟล์ด้วยเงื่อนไข `if` เช่นเดียวกับการเขียนข้อมูลบันทึกลงในไฟล์ จากนั้นรับค่าจากการอ่านไฟล์โดยคำสั่ง `fread(all, num*sizeof(Employee), 1, fp)`; คำสั่งนี้จะเป็นการอ่านไฟล์ข้อมูลตามขนาดที่กำหนดและตามจำนวนที่กำหนดในไฟล์ที่ fp ชี้ ในที่นี้จะเป็นขนาดของ Employee ที่ใช้ในการเก็บของข้อมูลกับจำนวนพนักงานที่รับค่ามาในตัวแปร num หลังจากนั้นจะทำการวนลูปภายใต้เงื่อนไข `for (i = 0; i < num; i++)` แล้วแสดงค่าข้อความชื่อพนักงานในตัวแปร `(all+i)->name` และค่าเงินเดือนในตัวแปร `(all+i)->salary` วนลูปจนครบเงื่อนไขและใช้คำสั่งปิดไฟล์ `fclose(fp)`; ต่อมาทำการแสดงค่าผลรวมเงินเดือนของพนักงานจากการเรียกใช้ฟังก์ชัน `total_salary(all, num)` ซึ่งเป็นการส่งค่าของ all และค่าจำนวนพนักงาน num ไปยังฟังก์ชัน `total_salary` และทำการคืนหน่วยความจำของ all ด้วยคำสั่ง `free(all)`;

ฟังก์ชัน `float total_salary(Employee all[], int len)` เป็นฟังก์ชันที่ได้รับค่า all เก็บในตัวแปร array all และค่าจำนวนพนักงาน num เก็บในตัวแปร len แล้วประกาศตัวแปรประเภท `int i` เพื่อวนลูปและตัวแปรประเภท `float sum` เพื่อคำนวณผลรวมของเงินเดือน ทำการวนลูปภายใต้เงื่อนไข `for (i=0; i<len; i++)` หลังจากนั้นจะใช้คำสั่ง `sum+= all[i].salary`; เพื่อคำนวณหาผลรวมเงินเดือนของพนักงาน ทำต่อไปเรื่อย ๆ จนครบเงื่อนไขของลูป และคืนค่าของ sum ไปให้ฟังก์ชัน `main ()`

ความรู้จากการทำ Lab Array of Employee : ไฟล์ binary เป็นไฟล์ที่ใช้งานกับโครงสร้างข้อมูล data structure โดยจะมีคำสั่ง `fwrite` ในการบันทึกข้อมูลลงในไฟล์และคำสั่ง `fread` ในการอ่านข้อมูลจากไฟล์ซึ่งคำสั่ง `fwrite` และ `fread` ต้องกำหนดขนาดและจำนวนอย่างเหมาะสมเพื่อให้ค่าออกมาถูกต้องและตรงตามที่ต้องการ