

ใบงานการทดลองที่ 16
เรื่อง เทคนิคการค้นหาและการเรียงข้อมูลภายในตัวแปรอาเรย์

1. จุดประสงค์ทั่วไป

- 3.1. รู้และเข้าใจแนวทางการประยุกต์การเขียนโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

- 3.1.10. ออกแบบแนวทางการค้นหาข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.11. วิเคราะห์แนวทางการค้นหาข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.12. วางหลักการโครงสร้างการค้นหาข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.13. ฝึกหัดและทดลองการค้นหาข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.14. แก้ไขและประยุกต์การค้นหาข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.15. ออกแบบแนวทางการจัดเรียงข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.16. วิเคราะห์แนวทางการจัดเรียงข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.17. วางหลักการโครงสร้างการจัดเรียงข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.18. ฝึกหัดและทดลองการจัดเรียงข้อมูลภายในตัวแปรชนิดอาเรย์
3.1.19. แก้ไขและประยุกต์การจัดเรียงข้อมูลภายในตัวแปรชนิดอาเรย์

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

- 4.1. จงบอกและอธิบายหลักการทำงานของการทำงานการค้นหาข้อมูลแบบเรียงลำดับ

การค้นหาลำดับ เรียงไว้ก่อน 9 ตัวคือ การค้นหาแบบเรียงลำดับ เช่น อักษร เลข รหัสผ่าน
หรือการเรียงใน Array หรือ Structure

- 4.2. จงวาดภาพประกอบการทำงานในข้อที่ 4.1 มาอย่างละเอียด

ภาพประกอบการทำงานอย่างละเอียด	
<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> $i = 0 \quad i \rightarrow s$ $find > nx$ </div> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> $a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5$ </div> <div style="margin-top: 10px;"> $if \ a[i] \geq nx$ $\quad \quad \quad output \ a[i]$ </div>	

4.3. จงบอกและอธิบายหลักการทำงานของการทำงานของเรียงลำดับข้อมูลแบบเลือก

การเลือกข้อมูล โดยดูว่าข้อมูลใดมีค่ามากที่สุดหรือน้อย แล้วนำค่าที่มากที่สุดหรือน้อยมาแลกเปลี่ยนกับค่าที่ตำแหน่งแรกสุดหรือสุดท้ายของ array โดยใช้ pointer เพื่อเก็บตำแหน่ง

4.4. จงวาดภาพประกอบการทำงานของขั้นตอนที่ 4.3 มาอย่างละเอียด

ภาพประกอบการทำงานอย่างละเอียด

$i = 0$ $j \rightarrow 5$

find max

a_1	a_2	a_3	a_4	a_5
-------	-------	-------	-------	-------

Start \rightarrow if $a[i] \geq \text{max}$
output $a[i]$

$i = 0$ $j \rightarrow 5$

find max

a_1	a_2	a_3	a_4	a_5
-------	-------	-------	-------	-------

Start \rightarrow if $a[i] \geq \text{max}$
output $a[i]$

$i = 0$ $j \rightarrow 5$

find max

a_5	a_4	a_3	a_2	a_1
-------	-------	-------	-------	-------

Start \rightarrow if $a[i] \geq \text{max}$
output $a[i]$

5. ลำดับชั้นการปฏิบัติงาน

5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.1.1. จากไฟล์ข้อมูล Salary.txt จงเขียนโปรแกรมเพื่อกระทำการดังต่อไปนี้

- เรียงลำดับชื่อของคนที่มียามามากที่สุดไปยังคนที่มีอายุน้อยที่สุดในบริษัท (เฉพาะ 5 คนแรก)
- เรียงลำดับคนชื่อของคนที่มียามเดือนน้อยที่สุด ไปยังคนที่มีเงินเดือนมากที่สุด (เฉพาะ 5 คนแรก)
- กรอกรายๆ เพื่อทำการค้นหาชื่อของคนที่มียามดังกล่าว

Salary.txt

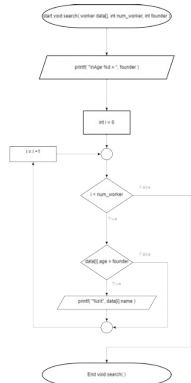
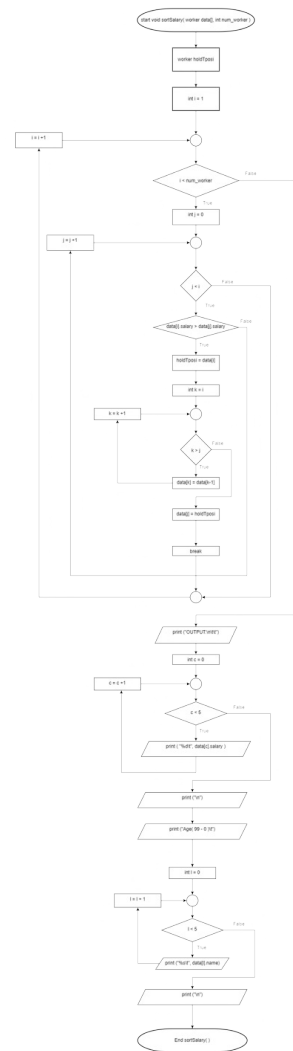
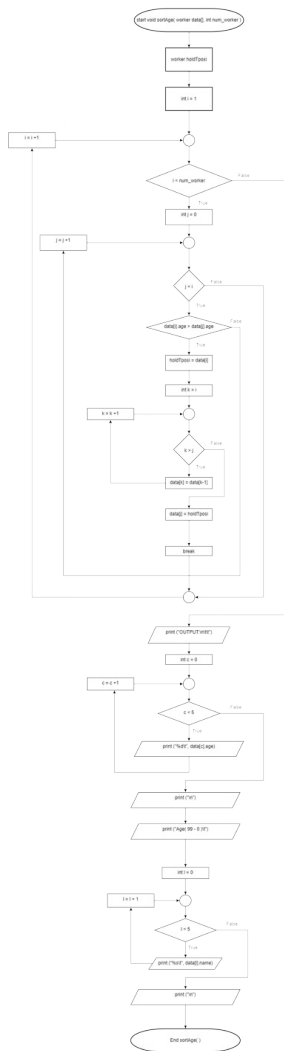
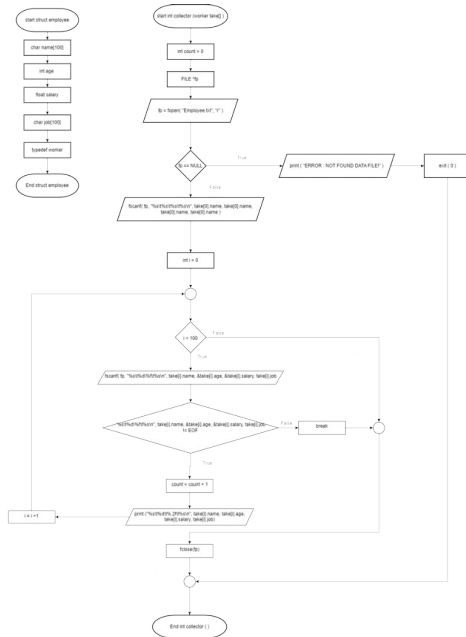
Name	Age	Salary(Bath)	Job
Kim	25	31,250	Advisor
Pong	32	45,000	Programmer
Som	25	23,000	Support
Aoy	24	33,250	Advisor
Ying	26	34,550	Programmer
DJ	29	21,150	HR
Yot	28	50,000	Founder
Pot	31	24,500	HR
Vip	25	25,450	Programmer
A	21	22,250	Support

Test case 1

Input	Find Age : 29					
Output	32	31	29	28	26	
	Age (99 - 0) : Pong	Pot	DJ	Yot	Ying	
		21150	22250	23000	24500	25450
	Salary(99999 - 0) :	DJ	A	Som	Pot	Vip
	Age 29 = DJ					

5.1.2. จงเขียนผังงาน

ผังงาน



5.1.3. จงเขียนโค้ดโปรแกรม

โค้ดโปรแกรม

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct employee {
    char name[100];
    int age;
    float salary;
    char job[100];
};typedef worker;

int collector( worker [] );
void sortAge( worker [], int );
void sortSalary( worker [], int );
void search( worker [], int, int );

int main() {
    FILE *fp;
    int countL;
    int searcher;
    worker employee[100];
    countL = collector(employee);
    printf("-----\n");
    printf("Find Age : ");
    scanf( "%d", &searcher );
    sortAge( employee, countL );
    sortSalary( employee, countL );
    search( employee, countL, searcher );
}

int collector( worker take[] ) {
    int count = 0;
    FILE *fp;
    fp = fopen( "Employee.txt", "r" );
    if (fp == NULL ){
        printf( "ERROR : NOT FOUND DATA FILE!" );
        exit( 0 );
    }
    fscanf( fp, "%s\t%s\t%s\t%s\n", take[0].name, take[0].name,
    take[0].name, take[0].name );
    for (int i = 0 ; i < 100 ; i++){
        if (fscanf( fp, "%s\t%d\t%f\t%s\n", take[i].name, &take[i].age,
        &take[i].salary, take[i].job ) != EOF) {
            count++;
            printf("%s\t%d\t%.2f\t%s\n", take[i].name, take[i].age,
            take[i].salary, take[i].job);
        }
        else{
            break ;
        }
    }
    fclose(fp);
    return count ;
}

void sortAge( worker data[], int num_worker ) {
    worker holdTposi;
    for (int i = 1; i < num_worker ; i++) {
        for (int j = 0 ; j < i ; j++) {
            if( data[j].age > data[i].age){
                holdTposi = data[i];
                for( int k = i ; k > j ; k--){
                    data[k] = data[k-1];
                }
                data[j] = holdTposi;
                break ;
            }
        }
    }
    printf("OUTPUT:\n\t\t");
    for (int c = 0 ; c < 5 ; c++){
        printf("%dt", data[c].age);
    }
    printf("\n");
    printf("Age( 99 - 0 )\t");
    for (int i = 0 ; i < 5 ; i++){
        printf("%st", data[i].name);
    }
    printf("\n");
}

void sortSalary( worker data[], int num_worker ) {
    worker holdTposi;
    for (int i = 1; i < num_worker ; i++) {
        for (int j = 0 ; j < i ; j++) {
            if( data[j].salary < data[i].salary){
                holdTposi = data[i];
                for( int k = i ; k > j ; k--){
                    data[k] = data[k-1];
                }
                data[j] = holdTposi;
                break ;
            }
        }
    }
    printf("OUTPUT:\n\t\t");
    for (int c = 0 ; c < 5 ; c++){
        printf("%.fv", data[c].salary);
    }
    printf("\n");
    printf("Salary( 99999 - 0 )\t");
    for (int i = 0 ; i < 5 ; i++){
        printf("%st", data[i].name);
    }
    printf("\n");
}

void search( worker data[], int num_worker, int founder ) {
    printf( "\nAge %d = ", founder );
    for ( int i = 0 ; i < num_worker ; i++) {
        if( data[i].age == founder ){
            printf( "%st", data[i].name );
        }
    }
}
```

6. สรุปผลการปฏิบัติงาน

nama: JPM Test Case Tools dan cara menggunakannya

Loop သည် ပုံသေ နံပါတ်များကို

7. คำถามทางการทดลอง

7.1. จรรยาบรรณควรระวังในการค้นหาข้อมูล

การคำนวณของ Profit & Loss ของบริษัท การคำนวณต้นทุน และ กำไร และ ขาดทุน
การคำนวณต้นทุน การคำนวณต้นทุน และ กำไร และ ขาดทุน

7.2. จรรยาบรรณควรระวังในการเรียงข้อมูล

7.3. จงบอกแนวทางการ Swap ข้อมูล

notu bajan: mis / 22 hr strepy any

7.4. จงระบุความเหมือน/แตกต่างในการเรียงข้อมูลระหว่างตัวแปรธรรมดาและตัวแปรโครงสร้าง
ข้อมูลมาอย่างละเอียด

