

ใบงานการทดลองที่ 16
เรื่อง เทคนิคการค้นหาและการเรียงข้อมูลภายในตัวแปรอาร์เรย์

1. จุดประสงค์ทั่วไป

3.1. รู้และเข้าใจแนวทางการประยุกต์การเขียนโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

- 3.1.10. ออกแบบแนวทางการค้นหาข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.11. วิเคราะห์แนวทางการค้นหาข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.12. วางหลักการโครงสร้างการค้นหาข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.13. ฝึกหัดและทดลองการค้นหาข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.14. แก้อัปเดตการค้นหาข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.15. ออกแบบแนวทางการจัดเรียงข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.16. วิเคราะห์แนวทางการจัดเรียงข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.17. วางหลักการโครงสร้างการจัดเรียงข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.18. ฝึกหัดและทดลองการจัดเรียงข้อมูลภายในตัวแปรชนิดอาร์เรย์
- 3.1.19. แก้อัปเดตการจัดเรียงข้อมูลภายในตัวแปรชนิดอาร์เรย์

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

4.1. จงบอกและอธิบายหลักการทำงานของการทำงานของการค้นหาข้อมูลแบบเรียงลำดับ

การค้นหาข้อมูลแบบเรียงลำดับ เป็นการค้นหาข้อมูลที่อยู่ในอาร์เรย์ โดยเรียงข้อมูลไว้ก่อนแล้วจึงไปค้นหาข้อมูลที่ต้องการในอาร์เรย์นั้น

int Num[10] = {53, 16, 42, 49, 88, 19, 24, 37, 45, 13};

4.2. จงวาดภาพประกอบการทำงานในข้อที่ 4.1 มาอย่างละเอียด

ภาพประกอบการทำงานอย่างละเอียด										
	0	1	2	3	4	5	6	7	8	9
int Num[10]	53	16	42	49	88	19	24	37	45	13

! - - - [53] == [24]
Num[0] Find.

4.3. จงบอกและอธิบายหลักการการทำงานของเรียงลำดับข้อมูลแบบเลือก
 ระบบการเรียงข้อมูลแบบทัวไปทัวจก. การทำงานจะมีขั้นตอนค่าเลขที่ลำดับที่เล็ก
 จนกระทั่งสุดท้าย จนพบค่าที่น้อยที่สุดที่สุด แล้วทำการสลับตำแหน่ง
 ไป

4.4. จงวาดภาพประกอบการทำงานในข้อที่ 4.3 มาอย่างละเอียด

ภาพประกอบการทำงานอย่างละเอียด

44 , 33 , 11 , 55 , 77 , 60

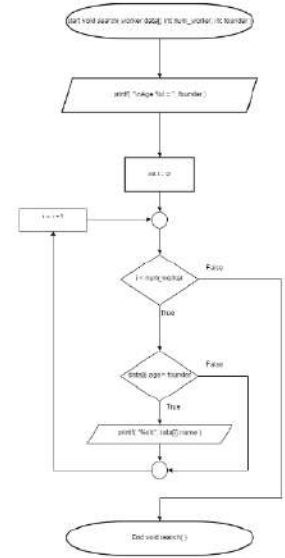
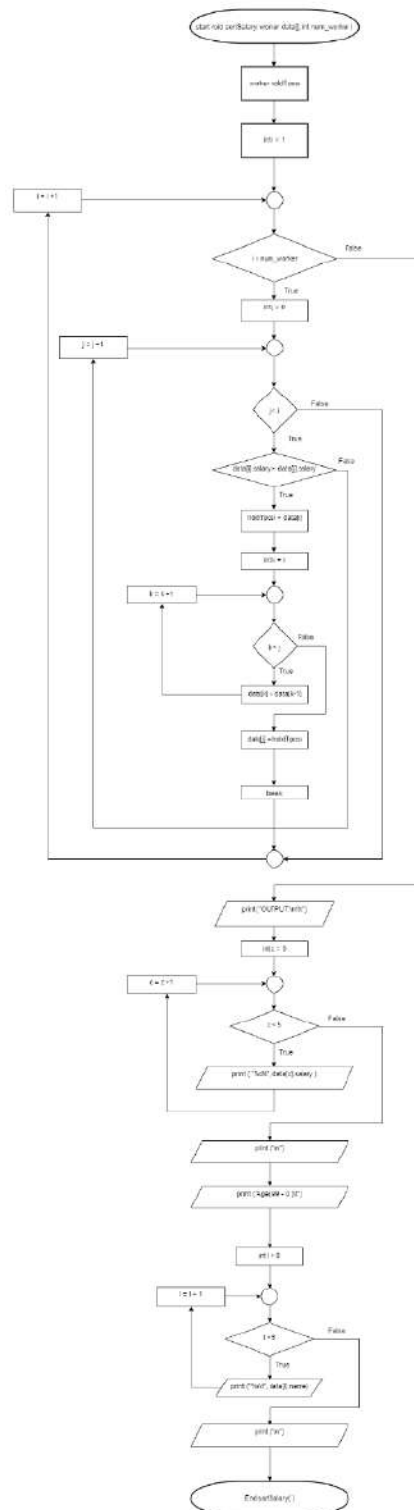
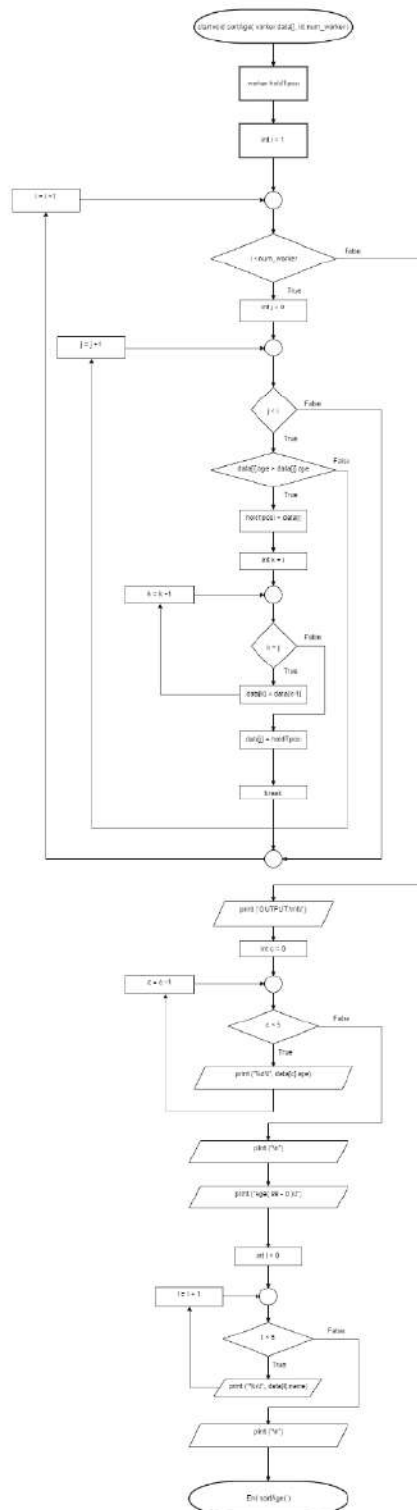
44	33	11	55	77	60
----	----	----	----	----	----

↓ ค้นหา

11	33	44	55	77	60
----	---------------	---------------	----	----	----

↓ ค้นหา

11	33	44	60	77	55
----	----	----	----	----	----



```

#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 ) ;

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[i].age > data[j].age){
                holdTposi = data[i] ;
                for ( int k = i ; k > j ; k--){
                    data[k] = data[k-1] ;
                }
                data[j] = holdTposi ;
                break ;
            }
        }
    }

    printf("OUTPUT:\n\n");
    for (int c = 0 ; c < 5 ; c++){
        printf("%d\t", data[c].age) ;
    }
}

```

```

        printf("\n") ;
        printf("Age( 99 - 0 )\t") ;
        for (int l = 0 ; l < 5 ; l++){
            printf("%s\t", data[l].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[i].salary < data[j].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\t") ;
    for (int c = 0 ; c < 5 ; c++){
        printf("%.f\t", data[c].salary) ;
    }
    printf("\n") ;
    printf("Salary( 99999 - 0 )\t") ;
    for (int l = 0 ; l < 5 ; l++){
        printf("%s\t", data[l].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( "%s\t", data[i].name ) ;  
        }  
    }  
}
```


7. คำถามทางการทดลอง

7.1. จงระบุข้อควรระวังในการค้นหาข้อมูล

ควรเป็นโพธิ์ขาวข้อมูลบน 14 ที่นิยมใช้ในการจัดข้อมูล

7.2. จงระบุข้อควรระวังในการเรียงข้อมูล

ควรระบุค่ามากที่สุดกับน้อยให้ล้นค่าที่ชัดเจน

7.3. จงบอกแนวทางการ Swap ข้อมูล

คือ การสลับข้อมูลที่มีค่าที่อยู่มากกว่า 49

7.4. จงระบุความเหมือน/แตกต่างในการเรียงข้อมูลระหว่างตัวแปรธรรมดาและตัวแปรโครงสร้าง

ข้อมูลมาอย่างละเอียด

เหมือนกันยกเว้นที่ต่างกันกับข้อมูลในข้อนี้ยกเว้นค่าตัวแปรปกติที่ไม่แน่นอนกัน