

ใบงานการทดลองที่ 14
เรื่อง เทคนิคการหาค่ามากที่สุดและน้อยที่สุด

1. จุดประสงค์ทั่วไป

3.1. รู้และเข้าใจแนวทางการประยุกต์การเขียนโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

3.1.1. ออกแบบแนวทางการแก้ไขปัญหาเพื่อหาค่ามากที่สุดและน้อยที่สุด

3.1.2. วิเคราะห์แนวทางการหาค่ามากที่สุดและน้อยที่สุด

3.1.3. วางหลักการโครงสร้างการหาค่ามากที่สุดและน้อยที่สุด

3.1.4. ฝึกหัดและทดลองการหาค่ามากที่สุดและน้อยที่สุด

3. เครื่องมือและอุปกรณ์

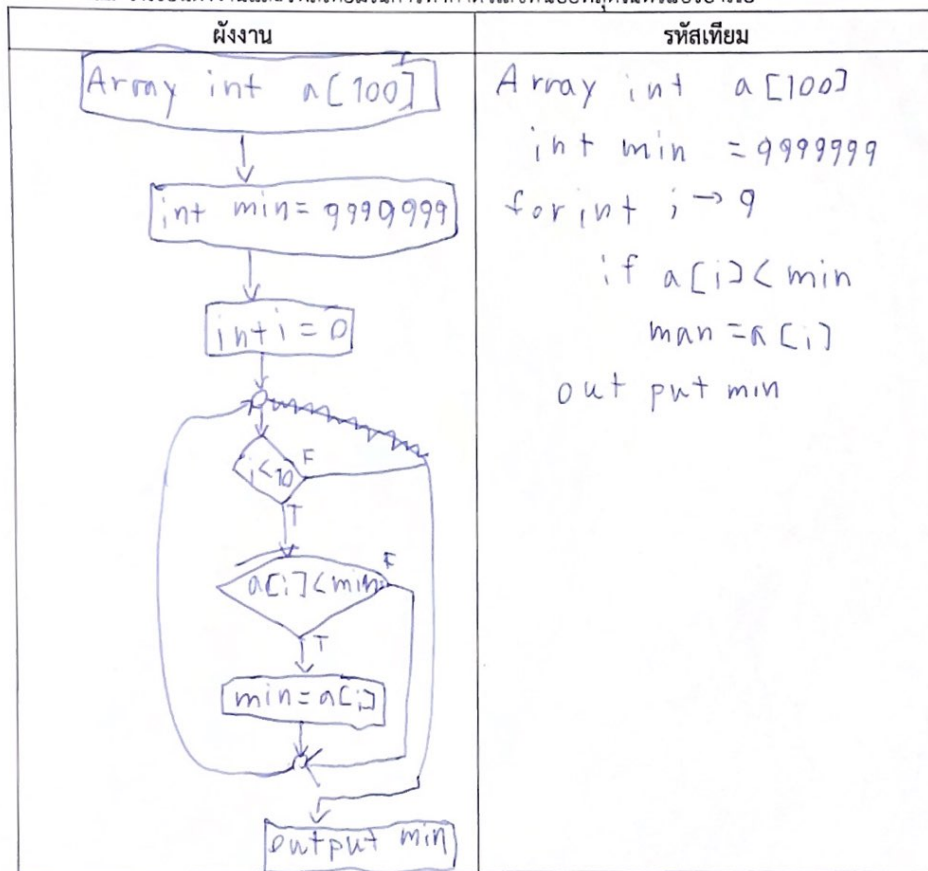
เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

4.1. จงเขียนผังงานและรหัสเทียมในการหาค่าตัวเลขที่มากที่สุดในตัวแปรอาเรย์

ผังงาน	รหัสเทียม
<pre> graph TD A[Array int a [100]] --> B[int max = -9999999] B --> C[int i = 0] C --> D{i < 10} D -- F --> H[output max] D -- T --> E{a[i] > max} E -- F --> H E -- T --> F[max = a[i]] F --> D </pre>	<pre> Array int a[100] int max = -9999999 for int i = 0 to 9 if a[i] > max max = a[i] output max </pre>

4.2. จงเขียนผังงานและรหัสเทียมในการหาค่าตัวเลขที่น้อยที่สุดในตัวแปรอาร์เรย์



4.3. จงยกตัวอย่างการเรียกใช้งานคำสั่งในฟังก์ชันหลักเพื่อหาค่าที่มากที่สุดในอาร์เรย์

```

for (i = 0; i < count; i++) {
    if (data[i] > min) {
        max = data[i];
    }
}

```

4.4. จงยกตัวอย่างการเรียกใช้งานคำสั่งในฟังก์ชันหลักเพื่อหาค่าที่น้อยที่สุดในอาร์เรย์

```

for (i = 0; i < count; i++) {
    if (data[i] < min) {
        min = data[i];
    }
}

```

5. ลำดับชั้นการปฏิบัติงาน

5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.1.1. จงบันทึกข้อมูลต่อไปนี้ลงในโครงสร้างข้อมูลแบบอาเรย์

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

และจงสร้างฟังก์ชันการทำงานเพื่อหาคนที่มีอายุเยอะที่สุดในบริษัทนี้ พร้อมทั้งหาผู้ที่มีเงินเดือนน้อยที่สุดในบริษัทนี้

Test case

Input	
Output	<p>Oldest = Pong (32) Programmer</p> <p>Low Salary = Som (25) 23,000 Support</p>

5.1.2. จงเขียนผังงาน

ผลงาน

5.1.3. จงเขียนโค้ดโปรแกรม

โค้ดโปรแกรม

```

1  #include<stdio.h>
2  #include<string.h>
3  struct Employee{
4      char Name[100] ;
5      int age ;
6      float Salary ;
7      char Job[100] ;
8  }typedef emp ;
9
10 int main() {
11     emp worker[100] ;
12     int i ;
13     int oldest = -999999999 ;
14     char *oldest_name ;
15     char *oldest_job ;
16     float lowestSal = 999999999999 ;
17     char *lowestSal_name ;
18     int *lowestSal_age ;
19     char *lowestSal_job ;
20     char name[100][100] = {"Pong", "Sorn", "Aoy", "Ying", "Yot", "Pot", "Vop"} ;
21     int age[100] = {32, 25, 24, 26, 28, 31, 25} ;
22     float salary[100] = { 45000, 23000, 33250, 34550, 50000, 24500, 25450 } ;
23     char job[100][100] = { "Programmer", "Support", "Advisor", "Programmer", "Founder", "HR", "Programmer"} ;
24     char holdername[100], holderjob[100] ;
25
26     for ( i = 0 ; i < 7 ; i++ ) {
27         for ( int j = 0 ; j < 100 ; j++ ) {
28             if ( name[i][j] == NULL ) {
29                 holdername[j] = '\0' ;
30                 break ;
31             }
32             else {
33                 holdername[j] = name[i][j] ;
34             }
35         }
36         for ( int k = 0 ; k < 100 ; k++ ) {
37             if ( job[i][k] == NULL ) {
38                 holderjob[k] = '\0' ;
39                 break ;
40             }
41             else {
42                 holderjob[k] = job[i][k] ;
43             }
44         }
45         strcpy( worker[i].Name, holdername ) ;
46         strcpy( worker[i].Job, holderjob ) ;
47         holdername[0] = '\0' ;
48         holderjob[0] = '\0' ;
49         worker[i].age = age[i] ;
50         worker[i].Salary = salary[i] ;
51     }
52     for ( int j = 0 ; j < i ; j++ ) {
53         if( worker[j].age > oldest ) {
54             oldest = worker[j].age ;
55             oldest_name = worker[j].Name ;
56             oldest_job = worker[j].Job ;
57         }
58         if( worker[j].Salary < lowestSal ){
59             lowestSal = worker[j].Salary ;
60             lowestSal_name = worker[j].Name ;
61             lowestSal_age = &worker[j].age ;
62             lowestSal_job = worker[j].Job ;
63         }
64     }
65     printf("Oldest = %s (%d) %s ", oldest_name, oldest, oldest_job) ;
66     printf("\nLowest Salary = %s (%d) %.f %s ", lowestSal_name, *lowestSal_age, lowestSal, oldest_job ) ;
67 }

```


6. สรุปผลการปฏิบัติงาน

จาก 115 ทดสอบ พบว่า สามารถรับจำนวนของฟังก์ชันที่ถูกเก็บไว้ใน structure
ตัวเก็บนั้นได้จนจบ เว้นแต่ถ้าหากใส่ค่าที่เกินกว่าที่ structure ให้อยู่ for และ ไม่ให้ if เป็นเรื่องใหญ่

7. คำถามทางการทดลอง

7.1. จงอธิบายเหตุผลของการกำหนดค่าเริ่มต้นตัวแปร $\text{max} = -999999$

เพื่อให้เป็นค่าที่ต่ำที่สุดที่เป็นไปได้ในกรณีแรก

7.2. จงระบุความแตกต่างในการหาค่ามากที่สุด/น้อยที่สุดของตัวแปรอาเรย์แบบธรรมดาและตัวแปร
อาเรย์ของโครงสร้างข้อมูล

ความแตกต่างคือ ตัวแปร array ของโครงสร้างข้อมูล จะรับค่าที่น้อยที่สุด
และค่าแน่นอนได้ แต่ array ธรรมดาจะรับค่าที่น้อยที่สุด
อย่างไม่แน่นอน