

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     Start([start]) --&gt; Init["int array[] max &lt;- -99999 i &lt;- 0"]     Init --&gt; LoopStart(( ))     LoopStart --&gt; Cond1{"i &lt; n?"}     Cond1 -- f --&gt; End([End])     Cond1 -- t --&gt; Cond2{"array[i] &gt; max?"}     Cond2 -- f --&gt; IncI["i &lt;- i+1"]     Cond2 -- t --&gt; AssignMax["max &lt;- array[i]"]     AssignMax --&gt; IncI     IncI --&gt; LoopStart     </pre>	<pre> set int array[] max = -99999 i = 0 for i &lt; n     if array[i] &gt; max         max = array[i]     i = i+1 End </pre>



4.2. จงเขียนผังงานและรหัสเทียมในการหาค่าตัวเลขที่น้อยที่สุดในตัวแปรอาร์เรย์

ผังงาน	รหัสเทียม
<pre> graph TD     Start([start]) --&gt; Init[int array[] min &lt;- 99999 i &lt;- 0]     Init --&gt; LoopStart(( ))     LoopStart --&gt; Cond1{i &lt; n?}     Cond1 -- F --&gt; End([End])     Cond1 -- T --&gt; Cond2{array[i] &lt; min}     Cond2 -- F --&gt; Inc[i &lt;- i+1]     Cond2 -- T --&gt; AssignMin[min &lt;- array[i]]     AssignMin --&gt; Inc     Inc --&gt; LoopStart     </pre>	<pre> set array[] min = 99999 i = 0  for i &lt; n     if array[i] &gt; min         min = array[i]     i = i+1  End </pre>

4.3. จงยกตัวอย่างการเรียกใช้งานคำสั่งในฟังก์ชันหลักเพื่อหาค่าที่มากที่สุดในอาร์เรย์

```

for (i=0; i>n, i++){
    if (array[i] > max){
        max = array[i];
    } //end if
} //end for

```

4.4. จงยกตัวอย่างการเรียกใช้งานคำสั่งในฟังก์ชันหลักเพื่อหาค่าที่น้อยที่สุดในอาร์เรย์

```

for (i=0; i<n, i++){
    if (array[i] < min){
        min = array[i];
    } //end if
} //end for

```



## 5. ลำดับขั้นการปฏิบัติงาน

### 5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

#### 5.1.1. จงบันทึกข้อมูลต่อไปนี้ลงในโครงสร้างข้อมูลแบบอาเรย์

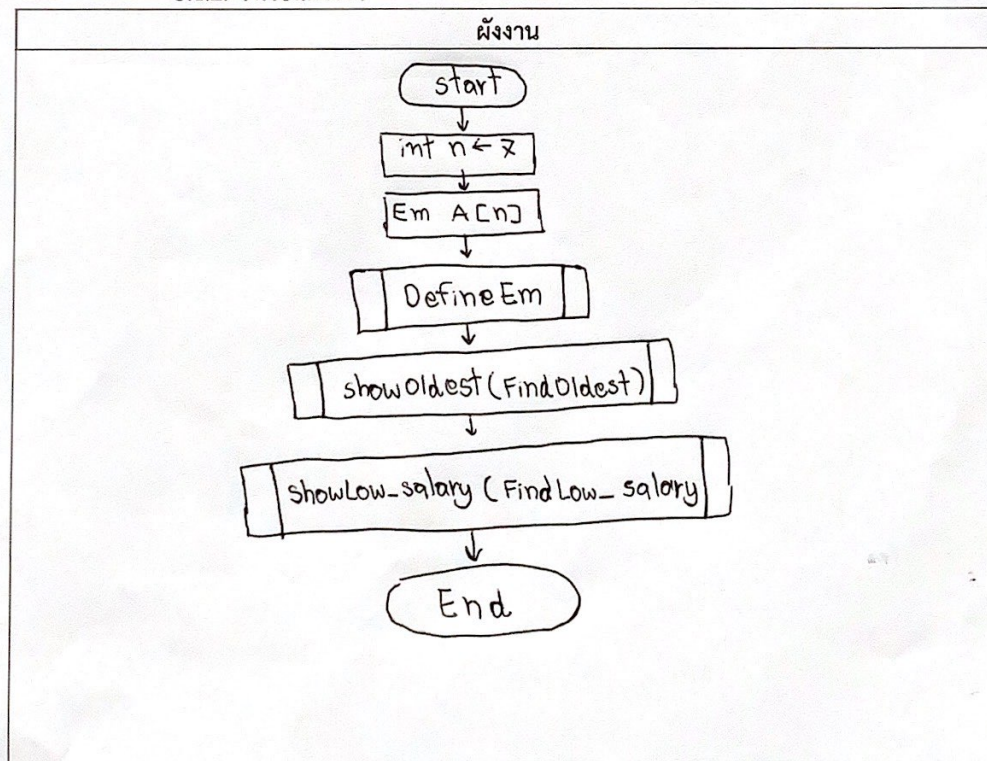
Name	Age	Salary(Bath)	Job
Pong	32	45,000	Programmer
Som	25	23,000	Support
Aoy	24	33,250	Advisor
Ying	26	34,550	Programmer
Yot	28	50,000	Founder
Pot	31	24,500	HR
Vip	25	25,450	Programmer

และจงสร้างฟังก์ชันการทำงานเพื่อหาคนที่มียายุเยะที่สุดในบริษัทนี้ พร้อมทั้งหาผู้ที่มีเงินเดือนน้อยที่สุดในบริษัทนี้

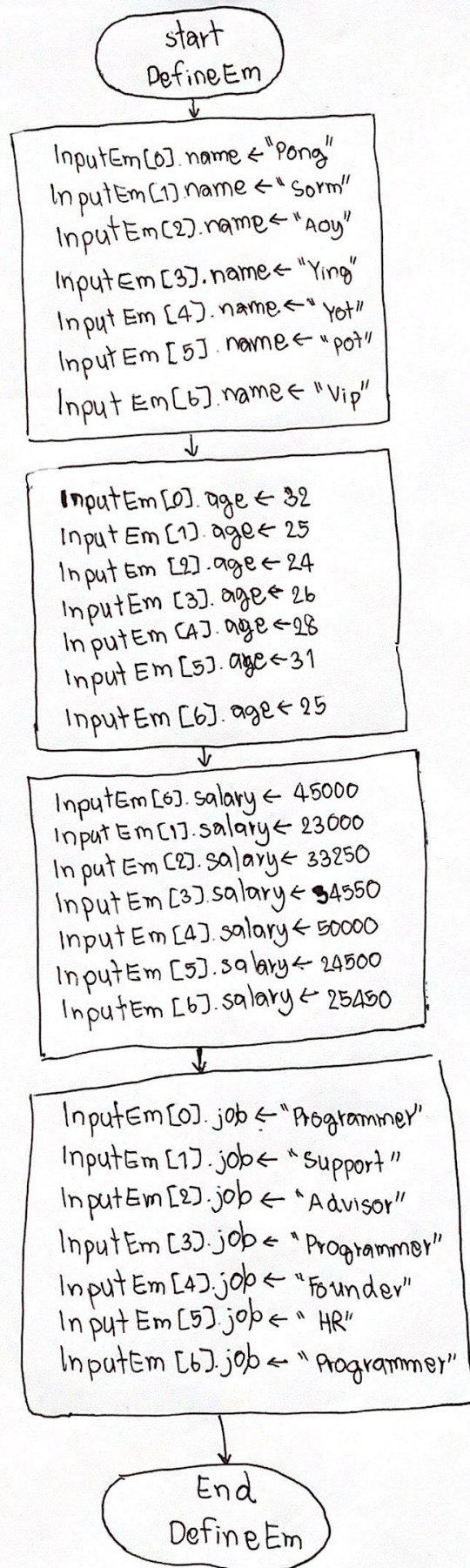
#### Test case

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

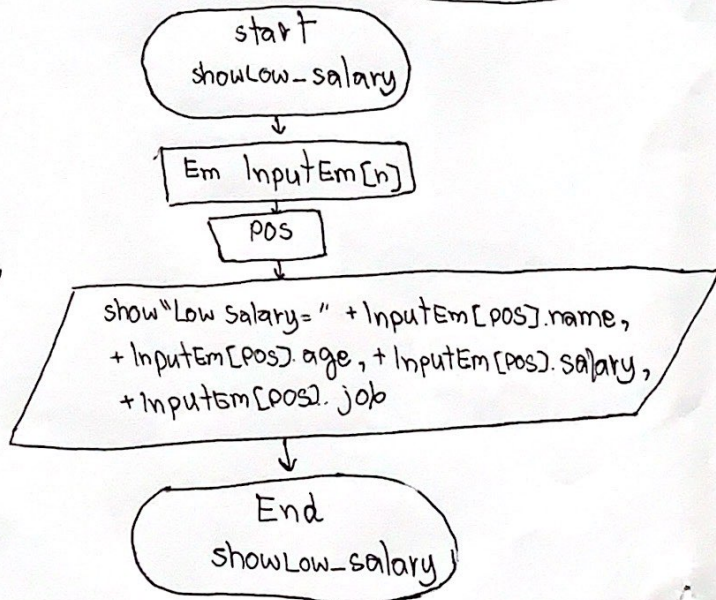
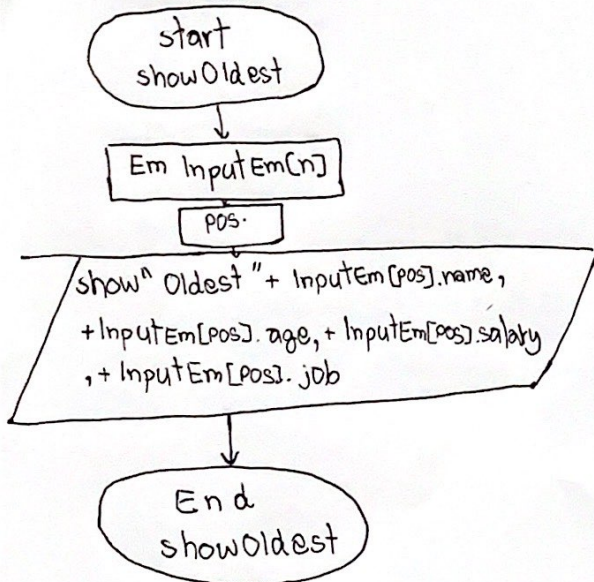
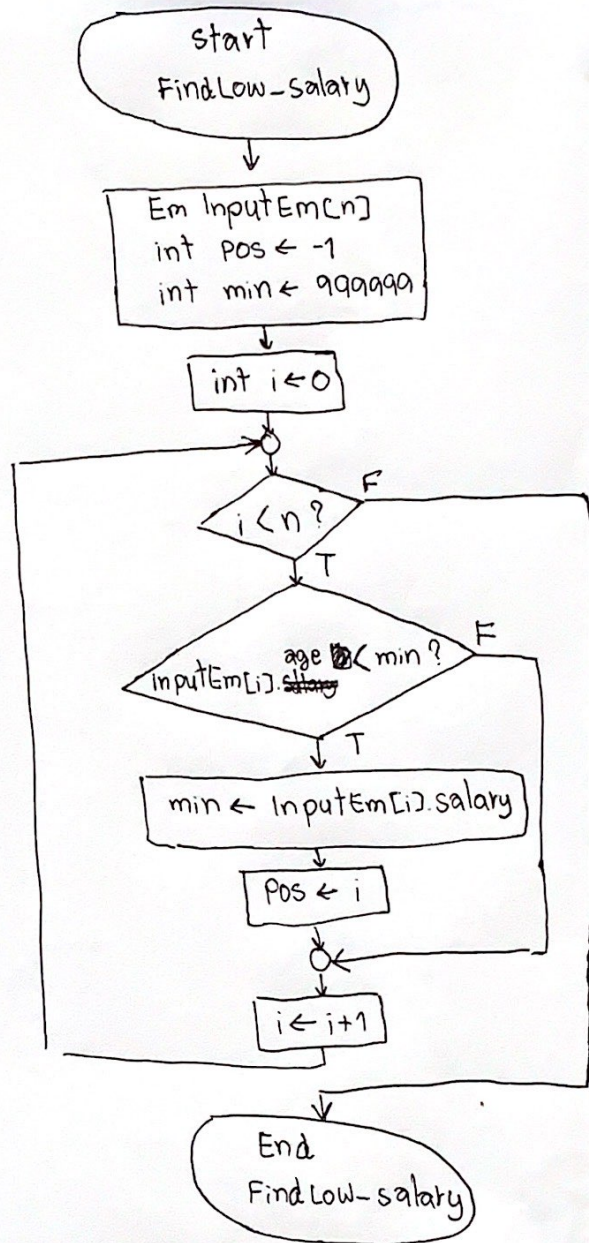
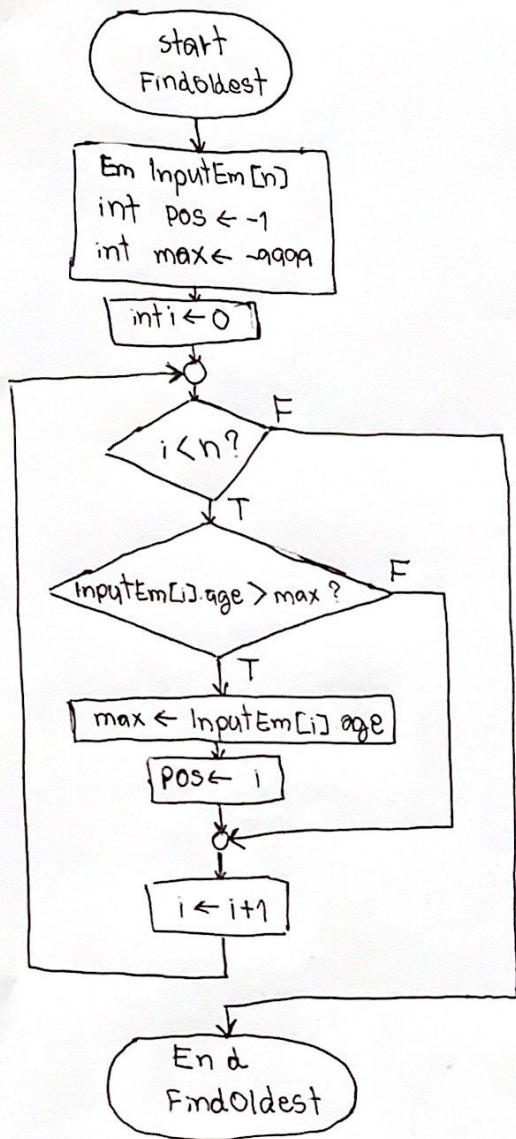
#### 5.1.2. จงเขียนผังงาน











### 5.1.3. จงเขียนโค้ดโปรแกรม

#### โค้ดโปรแกรม

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

struct Employees {
    char Name[ 20 ] ;
    int age ;
    int Salary ;
    char Job [ 20 ] ;
} typedef Em ;

void DefineEm( Em[] ) ;
int FindOldest(Em[] , int) ;
int FindLow_Salary(Em[] , int) ;
void ShowOldest(Em[] , int) ;
void ShowLow_Salary(Em[] , int) ;

int main() {
    int n = 7 ;
    Em A[ n ] ;
    DefineEm( A ) ;

    ShowOldest( A , FindOldest( A , n ) ) ;
    ShowLow_Salary( A , FindLow_Salary( A , n ) ) ;

    return 0 ;
} //end function main

void DefineEm( Em InputEm[] ){
    strcpy( InputEm[ 0 ].Name , "Pong" ) ;
    strcpy( InputEm[ 1 ].Name , "Sorm" ) ;
    strcpy( InputEm[ 2 ].Name , "Aoy" ) ;
    strcpy( InputEm[ 3 ].Name , "Ying" ) ;
    strcpy( InputEm[ 4 ].Name , "Yot" ) ;
    strcpy( InputEm[ 5 ].Name , "Pot" ) ;
    strcpy( InputEm[ 6 ].Name , "Vip" ) ;

    InputEm[ 0 ].age = 32 ;
    InputEm[ 1 ].age = 25 ;
    InputEm[ 2 ].age = 24 ;
    InputEm[ 3 ].age = 26 ;
    InputEm[ 4 ].age = 28 ;
    InputEm[ 5 ].age = 31 ;
    InputEm[ 6 ].age = 25 ;

    InputEm[ 0 ].Salary = 45000 ;
    InputEm[ 1 ].Salary = 23000 ;
    InputEm[ 2 ].Salary = 33250 ;
    InputEm[ 3 ].Salary = 34550 ;
    InputEm[ 4 ].Salary = 50000 ;
    InputEm[ 5 ].Salary = 24500 ;
    InputEm[ 6 ].Salary = 25450 ;

    strcpy( InputEm[ 0 ].Job , "Programmer" ) ;
    strcpy( InputEm[ 1 ].Job , "Support" ) ;
    strcpy( InputEm[ 2 ].Job , "Advisor" ) ;
    strcpy( InputEm[ 3 ].Job , "Programmer" ) ;
    strcpy( InputEm[ 4 ].Job , "Founder" ) ;
    strcpy( InputEm[ 5 ].Job , "HR" ) ;
    strcpy( InputEm[ 6 ].Job , "Programmer" ) ;
} //end function DefineEm

int FindOldest(Em InputEm[] , int n){
    int Pos = -1 ;
    int Max = -99999 ;
    for( int i = 0 ; i < n ; i++){
        if( InputEm[ i ].age > Max ){
            Max = InputEm[ i ].age ;
            Pos = i ;
        } //end if
    } // end for
    return Pos ;
} //end function FindOldest

int FindLow_Salary(Em InputEm[] , int n){
    int Pos = -1 ;
    int Min = 99999999 ;
    for( int i = 0 ; i < n ; i++){
        if( InputEm[ i ].age < Min ){
            Min = InputEm[ i ].Salary ;
            Pos = i ;
        } //end if
    } // end for
    return Pos ;
} //end function FindLow_Salary

void ShowOldest(Em InputEm[] , int Pos){
    printf( "Oldest = %s ( %2d ) %3d %10s\n" , InputEm[ Pos ].Name , InputEm[ Pos ].age , InputEm[ Pos ].Salary , InputEm[ Pos ].Job ) ;
} //end function ShowOldest

void ShowLow_Salary(Em InputEm[] , int Pos){
    printf( "Low Salary = %s ( %2d ) %3d %10s\n" , InputEm[ Pos ].Name , InputEm[ Pos ].age , InputEm[ Pos ].Salary , InputEm[ Pos ].Job ) ;
} //end function ShowLow_Salary
```



#### 6. สรุปผลการปฏิบัติงาน

สามารถหาค่าของตัวแปรที่เก็บไว้ใน struct มา  
เปรียบเทียบกับเงื่อนไขและเลือกค่ามากที่สุด หรือน้อยที่สุด โดยให้  
for วนมาหาค่าของตัวแปรโดยมี if เปรียบเทียบ

#### 7. คำถามทางการทดลอง

7.1. จงอธิบายเหตุผลของการกำหนดค่าเริ่มต้นตัวแปร max=-999999

เพื่อให้ค่าที่รับเข้ามาน้อยกว่าค่าเริ่มต้นในการเปรียบเทียบค่า  
ซึ่งมากกว่า ถ้ารับจำนวนยิ่งน้อย ยิ่งสามารถหา 100 เท่า  
ของค่ามากที่สุดได้มากกว่า

7.2. จงระบุความแตกต่างในการหาค่ามากที่สุด/น้อยที่สุดของตัวแปรอาเรย์แบบธรรมดาและตัวแปร  
อาเรย์ของโครงสร้างข้อมูล

การเข้าถึงคือ array ของ struct ต้องระบุตัวแปร  
ของตัว struct และตำแหน่งด้วย ซึ่งแตกต่างกับ array ทั่วไป  
ที่สามารถเข้าถึงตำแหน่งอย่างง่าย แต่ struct สามารถ  
ระบุ หรือระบุข้อมูลที่เกี่ยวข้องในตัวเองได้