

# ใบปฏิบัติการการเขียนโปรแกรมภาษาซี

## การทดลองที่ 6

### อาร์เรย์

#### วัตถุประสงค์

- เพื่อให้ผู้อ่านสามารถเขียน โปรแกรมการเก็บข้อมูลชนิดอาร์เรย์ได้
- เพื่อให้ผู้อ่านมีทักษะในการเขียน โปรแกรมด้วยอาร์เรย์ตั้งแต่หนึ่งมิติ และหลายมิติได้
- เพื่อให้ผู้อ่านสามารถเขียน โปรแกรมการ โดยนำทฤษฎีมาประยุกต์ใช้ ในการเขียน โปรแกรมได้

#### ความรู้พื้นฐาน

##### การทดลองการเขียนโปรแกรมภาษาซี

การจัดการข้อมูลที่ใช้ในการเขียน โปรแกรมคอมพิวเตอร์ ข้อมูลที่ใช้ในการเก็บ และนำมาประมวลผล ข้อมูลจะเป็นชนิดเดียวกันหรือกลุ่มเดียวกัน แต่ต้องจัดเก็บ ไว้ในชื่อของตัวแปรเดียวกัน ในการเรียกมาใช้งานในการประมวลผลต้องการเรียก ข้อมูลออกมาเป็นชุดเช่นกันนั้น การจัดการกับข้อมูลคอมพิวเตอร์ลักษณะนี้ สามารถดำเนินการได้ด้วย อาร์เรย์ ซึ่งสามารถสร้างชื่ออาร์เรย์ ทำงานโดยสามารถ

ทำการจัดเก็บข้อมูล และเรียกออกมาทำการประมวลผลได้อัตโนมัติ ด้วยคำสั่งด้านวนรอบ (ลูป) เช่น for() หรือ while() และ do/while ได้อย่างสะดวกรวดเร็ว เพื่อเกิดความชำนาญในการสร้างอาร์เรย์ มาใช้งานให้ผู้อ่านปฏิบัติ (Byron S. Gottfried, 1990) ดังนี้

โจทย์ที่ 1. โปรแกรมในการคำนวณหาค่าเบี่ยงเบนมาตรฐาน (deviation) เป็นค่าของแต่ละค่าที่มีความต่างกับค่าเฉลี่ยของจำนวนเลขทั้งหมด คือ ค่าเบี่ยงเบนมาตรฐาน เท่ากับ ค่าของเลขใดๆ ลบ ด้วยค่าเฉลี่ยทั้งหมด ซึ่งสามารถเขียนได้ตั้งโปรแกรมและผลการทำงานโปรแกรมดังด้านล่าง โดยให้ผู้อ่านโปรแกรม พร้อมเขียนผลการทำงานของโปรแกรม ตามค่าข้อมูลของผู้อ่านกำหนด พร้อมทั้งอธิบายการทำงานโปรแกรมมาพอเข้าใจ

```
#include<stdio.h>
#include<conio.h>
int x, count;
float store[20], avg, devi, sum=0;
void main()
{
clrscr();
/*Recieve value for X variable*/
printf("Enter numbers X for Deviation : ");
scanf("%d",&x);
```

ต่อจากด้านบน

```

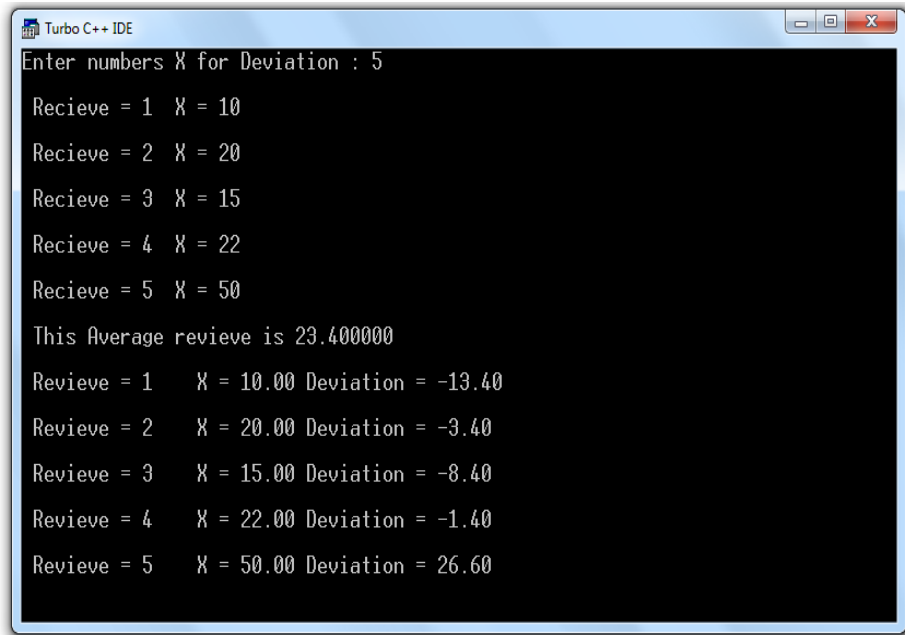
/*Caculate Summeried of X*/
    for(count=0; count < x; ++count)
    {
        printf("\n Recieve = %d X = ",count+1);
        scanf("%f",&store[count]);
        sum += store[count];
    }

/*caculate average*/
    avg = sum/x;
    printf("\n This Average reviewe is %f\n",avg);

/*caculate average and deviation value */
    for (count=0;count<x; ++count)
    {
        devi = store[count] - avg;
        printf("\n Reviewe = %d X = %3.2f Deviation = %3.2f
\n",count+1,store[count],devi);
    }
    getch();
}

```

## บันทึกผลการทำงานโปรแกรม



```

Turbo C++ IDE
Enter numbers X for Deviation : 5
Recieve = 1 X = 10
Recieve = 2 X = 20
Recieve = 3 X = 15
Recieve = 4 X = 22
Recieve = 5 X = 50
This Average review is 23.400000
Review = 1 X = 10.00 Deviation = -13.40
Review = 2 X = 20.00 Deviation = -3.40
Review = 3 X = 15.00 Deviation = -8.40
Review = 4 X = 22.00 Deviation = -1.40
Review = 5 X = 50.00 Deviation = 26.60
  
```

ให้ผู้อ่านเขียนผลกาทำงาน โปรแกรมของผู้อ่าน และอธิบายการทำงานของโปรแกรมพอเข้าใจ

โจทย์ที่ 2. โปรแกรมการรับตัวอักษร โดยนำค่าที่รับจากแป้นพิมพ์ของแต่ละชุด มาทำการเรียงลำดับตัวอักษร ซึ่งสามารถเขียนได้ดัง โปรแกรมและผลการทำงาน โปรแกรมดังด้านล่าง โดยให้ผู้อ่าน โปรแกรม พร้อมเขียนผลการทำงานของโปรแกรม ตามค่าข้อมูลของผู้อ่านกำหนด พร้อมทั้งอธิบายการทำงานของโปรแกรมมาพอเข้าใจ

ฟังก์ชัน `strcmpi( arg1,arg2)` เป็นฟังก์ชัน ถูกนิยามใน `string.h` ทำหน้าที่ในการเปรียบเทียบสตริง ระหว่าง `arg1` กับ `arg2` โดยผลการเปรียบเทียบให้ค่าเป็น ลบ ถ้าค่าสตริงตัวแรกน้อยกว่าสตริงตัวที่สอง ตามลำดับตัวอักษร (ASCII Table) 0 (ศูนย์) ค่าสตริงตัวแรกเท่ากับสตริงตัวที่สอง ตามลำดับตัวอักษร (ไม่ความแตกต่างตัวเล็กพิมพ์ใหญ่) บวก ถ้า ค่าสตริงตัวแรกมากกว่าสตริงตัวที่สอง ตามลำดับตัวอักษร (แสดงว่าสตริงตัวสองอยู่ก่อนสตริงตัวแรก)

```
#include<stdio.h> #include<conio.h>
#include<stdlib.h> #include<string.h>
void main()
{
    void sort(int x, char txt[][15]);
    int count,x = 0;
    char txt[20][15];
    clrscr();
    printf("Enter string for sorting\n");
    printf("Type \'end\' for finished\n");
    /*Receive string*/
    do {
        printf("String no %d : ",x+1);
        scanf("%s",&txt[x]);
    }
    while(strcmpi(txt[x++], "end"));
    sort(--x, txt);
}
```

ต่อจากด้านบน

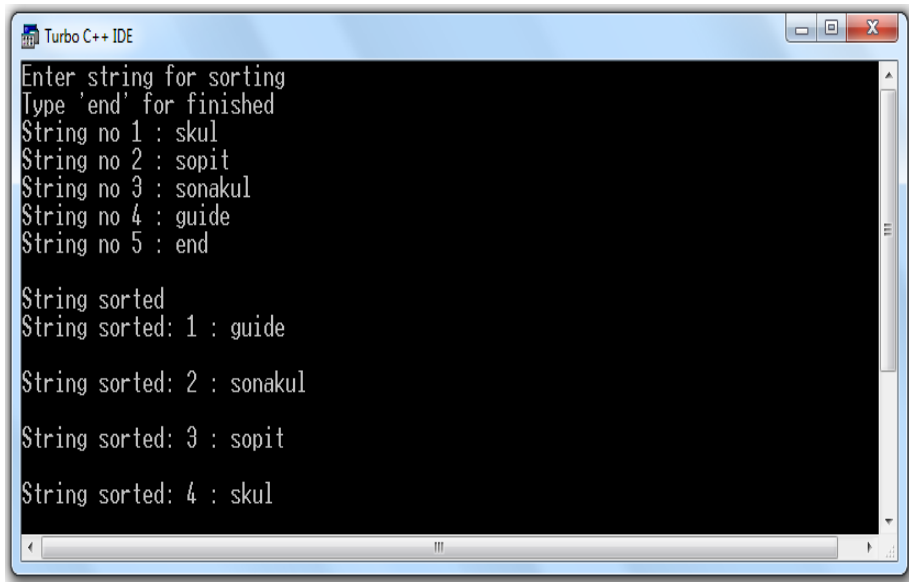
```

/*Show String sorted*/
printf("\nString sorted");
for (count=0; count < x; ++count)
{   printf("\nString sorted: %d : %s \n",count+1, txt[count]);   }
    getch();
}

/*String sorting by this code below*/
void sort(int n,char x[][15])
{   char temp[15];
    int i,item;
    for (item=0;item<n-1;++item)
        for(i=item+1;i<n;++i)
            {
                strcmpi(x[item],x[i]);
                if(x[i]>0) //change to string
                {
                    strcpy(temp, x[item]);
                    strcpy(x[item], x[i]);
                    strcpy(x[i], temp);
                }
            }
    return;
}

```

## ผลการทำงาน โปรแกรม



```
Turbo C++ IDE
Enter string for sorting
Type 'end' for finished
String no 1 : skul
String no 2 : sopit
String no 3 : sonakul
String no 4 : guide
String no 5 : end

String sorted
String sorted: 1 : guide
String sorted: 2 : sonakul
String sorted: 3 : sopit
String sorted: 4 : skul
```

ให้ผู้อ่านเขียนผลกาทำงาน โปรแกรมของผู้อ่าน และอธิบายการทำงานของโปรแกรมพอเข้าใจ

โจทย์ที่ 3. ให้ผู้อ่านเขียนโปรแกรมในลักษณะการโต้ตอบ เมื่อผู้ใช้โปรแกรมพิมพ์ชื่อประเทศเข้าไปในโปรแกรม ให้ตอบชื่อเมืองหลวงประเทศนั้นออกมา โปรแกรมจะทำงานเป็นวนรอบจะหยุดโปรแกรมเมื่อพิมพ์คำว่า end ในตัวโปรแกรมมีฐานข้อมูล โต้ตอบกับผู้ใช้งานโปรแกรมดังนี้

Thailand	Bangkok
U.S.A	Washington, D.C.
Canada	Ottawa
U.S.S.R	Moscow
China	Beijing
India	New Delhi
France	Paris
England	London
Italy	Rome
Japan	Tokyo

ให้ผู้อ่านเขียนผลกาทำงาน โปรแกรมของผู้อ่าน และอธิบายการทำงานขอโปรแกรมพอเข้าใจ

โจทย์ที่ 4. ให้ผู้อ่าน เขียนโปรแกรมในการแก้ปัญหาทางคณิตศาสตร์ตามกฎของ Boyle' Law คือ  $y = \sin 0.5t$  โดยกำหนดให้  $t$  มีค่าอยู่ระหว่าง 1 -60 บันทึกผลการทำงาน โปรแกรม และนำค่าที่ได้สร้างกราฟความสัมพันธ์ระหว่างค่า  $t$  (แกน  $x$ ) กับ ผลการคำนวณค่า  $y$  (แกน  $y$ ) นำผลการทำงานของโปรแกรมมาทำการสร้างกราฟความสัมพันธ์ระหว่าง  $t$  กับ  $y$  ( $y = \sin 0.5t$ )



## อุปกรณ์

1. เครื่องคอมพิวเตอร์
2. ระบบอินเทอร์เน็ต สำหรับการค้นคว้า
3. ตัวโปรแกรมภาษาซี หรือ
4. ตัวโปรแกรม CodeBlock
5. หน่วยความจำเคลื่อนที่ (Handy Drive, External Drive)

## วิธีการทดลอง

1. ให้ผู้อ่านเรียกโปรแกรมที่ใช้ในการเขียนโปรแกรมภาษาซี
  - 1.1 โปรแกรมภาษาซีจาก Turbo C
  - 1.2 โปรแกรม CodeBlock
2. ให้ผู้อ่านฝึกการ โปรแกรมที่ผู้อ่านได้เลือกทำการติดตั้งในเครื่องคอมพิวเตอร์
  - 2.1 ทดสอบการเขียนโค้ด
  - 2.2 ทดสอบการสั่งให้โปรแกรมทำการแปลคำสั่ง (Compiler)
  - 2.3 ทดลองทำการสั่งให้โปรแกรมเริ่มทำงาน (Run)
3. ให้ผู้อ่านทดสอบหาจุดผิดพลาดของโปรแกรมและทำการแก้ไขจุดผิดพลาดโดยยึดทฤษฎีมาเป็นหลักในการแก้ปัญหา
4. ให้ผู้อ่านได้ทดลองทำการป้อนตามที่เนื้อหาได้กำหนด ทั้งส่วนทฤษฎีและการทดลอง ให้ผลการทำงานโปรแกรม ตรงตามที่เนื้อหาที่กำหนด
5. ให้ผู้อ่านได้ทำการเขียนผลการทดลองตามกำหนด