ครั้งที่ 5

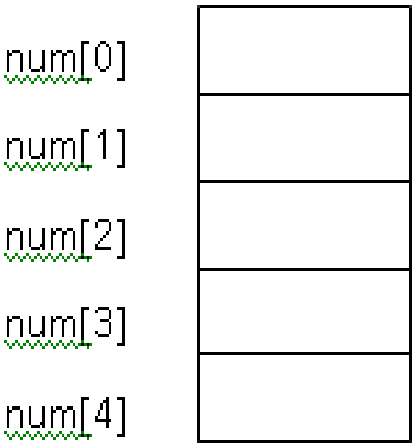
โครงสร้างข้อมูล

ชนิดอาเรย์ 1 มิติ

ข้อมูลชนิดอาเรย์

- เป็นโครงสร้างข้อมูลที่ยินยอมให้ตัวแปร 1 ตัว สามารถเก็บข้อมูลได้หลายค่าขึ้นอยู่กับความต้องการ

การประกาศตัวแปร



รูปแบบของการประกาศตัวแปรหนึ่งมิติ

dataType arrayName[intExp] ;

เช่น int num[5] ;

การประกาศตัวแปรนี้จะส่งผลให้ตัวแปร num สามารถเก็บข้อมูลได้ 5 จำนวนเป็นเลขจำนวนเต็ม อันประกอบด้วยตัวแปร

num[0] ,num[1] ,num[2] ,num[3] ,num[4]

Array Declaration

รูปแบบ: element-type array-name[size] ;

element-type array-name[size] = {initialization-list} ;

ตัวอย่าง: char myName[5] ;

float salaries[NUM_EMP] ;

char vowels[] = {'A' ,'E','I','O','U'} ;

Array Subscript

รูปแบบ: name[subscript] ;

ตัวอย่าง: NUM[3*I-2]

การปฏิบัติการ

กรณีที่กำหนดค่าเริ่มต้นให้แก่ตัวแปรอาเรย์

```
for (index=0 ; index<10 ; index++)
    num[index] = 0 ;
```

กำหนดค่าพร้อมกับประกาศตัวแปรชนิดอาเรย์ได้ดังนี้

```
int num[10] = {0} ;
```

กรณีที่อ่านข้อมูลมาเก็บในตัวแปรอาเรย์

```
for (index=0 ; index<10 ; index++)
    cin >> num[index]  ;
```

กรณีพิมพ์ข้อมูลที่เก็บในตัวแปรอาเรย์

```
for (index=0 ; index<10 ; index++)
    cout << num[index] << “   “ ;
```

การปฏิบัติการ

หาผลรวมและค่าเฉลี่ยของข้อมูลที่เก็บในตัวแปรอาเรย์

```
sum = 0 ;
 for (index = 0 ; index<10 ; index++)
        sum = sum + num[index] ;
average = sum /10.0 ;
```

หาค่าที่มากที่สุดของข้อมูลที่เก็บในอาเรย์

```
maxIndex = 0;
for (index = 1; index<10 ; index++)
    if (num[maxIndex] < num[index])          maxIndex = index ;
largestSale = num[maxIndex] ;
```

ตัวอย่าง โปรแกรมตัดเกรดนักศึกษา

- จงรับคะแนนของนักศึกษาจำนวน N คนทางแป้นพิมพ์ เก็บในตัวแปรอาเรย์ชื่อ STU ต่อจากนั้นอ่านข้อมูลจากตัวแปรอาเรย์ STU เพื่อหา
  - ค่าคะแนนที่มากที่สุด
  - คะแนนเฉลี่ย
  - เรียงลำดับคะแนนจากมากไปน้อย
  - จำนวนของนักศึกษาที่สอบได้เกรด P ,G และ F โดยใช้กฎเกณฑ์ของมหาวิทยาลัยรามคำแหง

ตัวอย่าง โปรแกรมตัดเกรดนักศึกษา

- เขียนโปรแกรมโดยมี 1 ฟังก์ชัน
- เขียนโปรแกรมโดยใช้เทคนิค Top-down

Selection Sort

- การเรียงลำดับข้อมูลในอาเรย์ โดยใช้หลักการในการค้นหาข้อมูลมาไว้ในตำแหน่งที่เหมาะสม

- สมมุติว่าข้อมูลก่อนเรียงลำดับมีดังนี้

[0]	[1]	[2]	[3]
74	45	83	16

สมมุติว่าข้อมูลก่อนเรียงลำดับมีดังนี้

[0]	[1]	[2]	[3]
74	45	83	16

หาให้ได้ว่าข้อมูลตัวที่น้อยที่สุดอยู่ในตำแหน่งไหน แล้วนำมาไว้เป็นลำดับแรก
ในที่นี้จะเห็นได้ว่า ข้อมูล 16 ในตำแหน่งที่ 3 น้อยที่สุด
ต้องนำมาไว้ในตำแหน่ง แรก
แต่ไม่ลืมที่จะนำข้อมูล 74 ไปไว้ในตำแหน่งที่เหมาะสม
ซึ่งเราก็ไม่รู้ว่าตำแหน่งไหน ทางที่ดีที่สุดคือ สลับข้อมูลกันระหว่างตำแหน่งที่ 0 กับ 3

[0]	[1]	[2]	[3]
16	45	83	74

[0]	[1]	[2]	[3]
16	45	83	74

กระทำแบบเดียวกันโดยสนใจเฉพาะข้อมูล
ตำแหน่งที่ 1 ถึง ตำแหน่งที่ 3 เท่านั้น
เมื่อพิจารณาจะเห็นว่า ข้อมูลตัวที่น้อยที่สุดคือ 45
อยู่ในตำแหน่งที่ 1 ซึ่งเป็นตำแหน่งที่เหมาะสมอยู่แล้ว
สำหรับขั้นตอนต่อไปเราจะพิจารณาข้อมูลที่เหลือ
คือตำแหน่งที่ 2 และตำแหน่งที่ 3
ซึ่งข้อมูลตัวที่น้อยอยู่ในตำแหน่งที่ 3
ต้องสลับค่าระหว่างข้อมูลในตำแหน่งที่ 2 และตำแหน่งที่ 3

[0]	[1]	[2]	[3]
16	45	74	83

อัลกอริทึมในการเรียงลำดับ

- กำหนดให้ i เป็นตัวแปรที่ระบุถึงตำแหน่งของข้อมูลที่มีค่าน้อยที่สุดในการทำงานแต่ละรอบ ในที่นี้ถ้า i เท่ากับ 0 ค่าที่น้อยที่สุดในการค้นหาจะอยู่ในตำแหน่งที่ 0 ค่าของ i จะเพิ่มขึ้นรอบละ 1 จนกระทั่งมีค่าเท่ากับ n-1 กำหนดให้ n คือจำนวนของข้อมูลทั้งหมดในอาเรย์
- การทำงานแต่ละรอบจะทำการค้นหาว่าข้อมูลที่น้อยที่สุดอยู่ในตำแหน่งใด ระบุถึงขอบเขตของข้อมูลที่ต้องการค้นหา
- เมื่อได้ตำแหน่งของข้อมูลตัวที่น้อยที่สุดแล้วต้องนำไปไว้ในตำแหน่งที่ถูกต้องโดยตรวจสอบกับค่าของ i ถ้าเหมือนกันแสดงว่าอยู่ในตำแหน่งที่ถูกต้องแล้ว แต่ถ้าไม่เหมือนกันต้องทำการสลับตำแหน่ง

ฟังก์ชันในการสลับค่า

```
void  exchange(int &  a  , int & b )
{
   int temp;
   temp      = a ;
   a   = b ;
   b   = temp ;
}
```

```
void selSort(int items[], int n)
{
        int minSub;

   for (int i = 0; i < n – 1; i++)
   {
        minSub = findIndexOfMin(items, i , n - 1);
        exchange (items[minSub], items[i]);
   }
}
```

อาร์กิวเมนต์อาเรย์ (Array Arguments)

- ส่งอาเรย์ผ่านฟังก์ชันได้ โดยเขียนชื่อแต่ไม่ต้องมี subscript
- ในลิสต์ของอากิวเมนต์ ของการเรียกฟังก์ชัน ซึ่งส่วนของ formal parameter list ของฟังก์ชันต้องมีการกำหนดพารามิเตอร์ให้สอดคล้องกับอากิวเมนต์ที่ส่งผ่านค่า โดยส่วนของพารามิเตอร์ต้องมีการกำหนด subscript เพื่อใช้อ้างถึงสมาชิกในอาเรย์

การส่งผ่านค่าที่เป็นอาเรย์

- ภาษา C++ นั้นยินยอมให้มีการส่งผ่านโครงสร้างข้อมูลชนิดอาเรย์แบบอ้างอิง โดยไม่มีการสร้างเนื้อที่ขึ้นใหม่แต่อ้างอิงใช้เนื้อที่ที่มีการเรียกใช้ที่สอดคล้องกัน
- การกำหนดฟังก์ชัน หรือต้นแบบฟังก์ชัน ใช้ [] ว่าง เพื่อบอกคอมไพเลอร์โดยไม่ต้องกำหนดถึงขนาดของอาเรย์
- คำเฉพาะ const ที่กำหนดหน้าพารามิเตอร์ แสดงให้ทราบว่าพารามิเตอร์นั้นไม่สามารถเปลี่ยนภายในฟังก์ชันได้ และถ้าพยายามแก้ไขข้อมูล คอมไพเลอร์จะแสดงข่าวสารของความผิดพลาดให้เราได้ทราบ
- สำหรับการเรียกใช้ (function call) เราเขียนชื่อ และอากิวเมนต์จริงโดยไม่ต้องใส่ [] หลังชื่อของตัวแปรอาเรย์

ฟังก์ชัน addArray

```
void addArray
    (int size,                      // IN: the size of the arrays
    const float a[],                // IN: the first array
    const float b[],                // IN: the second array
    float c[] )                     // OUT: result array
{
    for (int i  = 0; i  <  size;  i++ )
            c[i] = a[i] + b[i];
}
```

งาน

จงเขียนโปรแกรมรับเซตของตัวเลข 2 กลุ่มเก็บใน

ตัวแปรอาเรย์ A และ B ตามลำดับ จงหา

- A union B
- A intersection B
- A - B