C Programming

Lecture no. 7:

Arrays : ตัวแปรชุด : ตัวแปรแถวลำดับ

ทำไมต้องนิยามตัวแปรเป็นแถวลำดับ

- ตัวอย่าง : จงเขียนโปรแกรมสำหรับเก็บคะแนนนิสิตจำนวน 20 คน แล้วนำคะแนนทั้งหมดมารวมกัน
- จะต้องเขียนโปรแกรมดังนี้

```
int x01,x02,x03,x04,x05,x06,x07,x08,x09,x10;
int x11,x12,x13,x14,x15,x16,x17,x18,x19,x20;
int sum;
…
sum = x01+x02+x03+x04+x05+x06+x07;
sum = sum+x08+x09+x10+x11+x12+x13+x14;
sum = sum+x15+x16+x17+x18+x19+x20;
```

มีวิธีการที่ดีกว่านี้หรือไม่

อาเรย์ (Arrays)

- แถวลำดับเป็นโครงสร้างแบบ homogeneous ที่ประกอบด้วยสมาชิก (elements) ที่มีชนิด (type) เดียวกัน ในเบื้องต้นนี้จะกล่าวถึงอาร์เรย์มิติเดียว
- (Single-dimensional arrays) ซึ่งมีรูปแบบของการประกาศตัวแปรดังนี้

ชนิดข้อมูล ชื่อตัวแปร[ขนาดของอาเรย์]

ตัวอย่างการประกาศใช้ตัวแปรแบบอาเรย์

```
#define MAX 10
int x[MAX], y[MAX*2],z[MAX*3 + 5];
int v[4], w[4];
char s1[2], s2[1];
float f[10];
```

การใช้ตัวแปรอาเรย์ ให้พิจารณาประเภทของการใช้ดังนี้

- การใส่ค่าของสมาชิกเข้าอาร์เรย์

ตัวอย่าง

```
for (i=0;i < MAX; i++)
     a[i] = rand();
```

- การประมวลผลอาเรย์

```
for (i=0;i < MAX-1; i++) {
   t = a[i];
   a[i] = a[i+1];
   a[i+1] = t;
}
```

- การแสดงผลลัพท์จากอาร์เรย์

```
for (i=0;i < MAX; i++) {
    printf("%d\n",a[i]);
}
```

ศึกษาวิธีใช้อาเรย์ จากโปรแกรมตัวอย่างต่อไปนี้

```
#include <stdio.h>
#define MAX 10
…
int a[MAX];
```

ตัวแปรอาร์เรย์แบบ 1 มิติ

ตัวแปรแบบอาร์เรย์ ที่ใช้ในการเก็บข้อมูลโดยมีการอ้างอิงที่ใช้เลขดัชนีเพียงหนึ่งค่าจะเรียกว่า ตัวแปรแบบอาร์เรย์ 1 มิติ เช่น การเก็บข้อมูลคะแนน ที่อ้างอิงด้วยรหัสนิสิต

ตัวอย่าง

```
int age[10];
```

เป็นการประกาศตัวแปรแบบอาร์เรย์เพื่อเก็บข้อมูลอายุ

เมื่อต้องการเก็บข้อมูลอายุของนิสิตคนที่ 1 สามารถทำได้ดังนี้

```
age[0] = 18;
```

ข้อควรระวัง

ในการใช้งานอาร์เรย์จะมีการกำหนดขนาดและมีขอบของการใช้งานเช่น กำหนดตัวแปรขนาด 10 แต่อาจมีการทำงานถึงสมาชิกตัวที่ 12 โดยที่โปรแกรมจะไม่มีการแจ้งข้อผิดพลาดเมื่อทำการคอมไพล์แต่จะทำให้เกิดความผิดพลาดกับโปรแกรมได้ ซึ่งไม่เสมอไป หากตำแหน่งหน่วยความจำที่ถูกอ้างถึงมีค่าอยู่ แต่จะนำไปสู่ข้อผิดพลาดที่มากกว่านี้จึงควรระวัง ซึ่งเรียกความผิดพลาดที่มีการอ้างเกินขอบอันนี้ว่า error out of bounds

ตัวแปรอาร์เรย์แบบ 1 มิติ

การประกาศตัวแปรแบบอาร์เรย์สามารถทำได้คล้ายกับการประกาศตัวแปรอื่นๆ ทั่วไป โดยการกำหนดชื่อของตัวแปร ชนิดของข้อมูล และขนาดของข้อมูล ตัวอย่างเช่น

- `int grades[5];` /* Array ขนาด 5 ของ int */

  เป็นการประกาศตัวแปรแบบอาร์เรย์ 1 มิติชื่อ grades ให้เป็นข้อมูลแบบค่าตัวเลขที่เป็นจำนวนเต็มโดยมีขนาดเท่ากับ 5 อีลีเมนต์ ซึ่งการประกาศตัวแปรนี้จะจองหน่วยความจำเท่ากับ 2 byte * 5 = 10 byte

- `char codes[5];` /* Array ขนาด 5 ของ char */

  การประกาศตัวแปร codes เป็นข้อมูลแบบค่าตัวอักษรจะใช้หน่วยความจำเท่ากับ 1 byte * 5 = 5 byte

- `float prices[100];` /* Array ขนาด 100 ของ float */

  การประกาศตัวแปร price เป็นข้อมูลเลขทศนิยม จะใช้หน่วยความจำเท่ากับ 4 byte * 100 = 400 byte

ตัวแปรอาร์เรย์แบบ 1 มิติ

ในการอ้างถึงตัวแปรอาร์เรย์ ใดๆ จะต้องระบุด้วยชื่ออาร์เรย์และดัชนีซึ่งอยู่ภายในเครื่องหมาย [ ] ตัวแปรอาร์เรย์ตัวแรกจะมีดัชนี (index) เป็นศูนย์ ดังนั้น grades [2] หมายถึงตัวแปรอาร์เรย์ตัวที่ 3 ของ grades

ตัวอย่างการใช้งานตัวแปร grades ที่เป็นอาร์เรย์ของจำนวนเต็ม 5 ตัว

```
grades[0] = 98;
grades[1] = grades[0] - 11;
grades[2] = 2 * (grades[0] - 6);
grades[3] = 79;
grades[4] = (grades[2]+grades[3]-3)/2;
```

นั่นคือข้อมูลในอาร์เรย์ `grades [2]` คือ 184

สมมติว่าต้องการหาผลรวม

```
total = grades[0]+grades[1]+grades[2]+grades[3]+grades[4];
```

สามารถเปลี่ยนเป็น

```
total = 0;
for ( i = 0; i <= 4, ++i)
    total += grades[i];
```

จะเห็นว่าจะมีการเก็บค่าใน `grades` แต่ละสมาชิกตามลำดับเช่นกัน และสามารถใช้ `i` ซึ่งใช้เป็นตัวนับของลูป `for` ในขณะเดียวกันก็เป็นตัวชี้ให้กับอาร์เรย์ด้วย จะเห็นว่าจะมีประโยชน์มาก เมื่อมีการกระทำที่มีจำนวนครั้งมากกว่านี้

ตัวอย่าง กรณีที่ต้องการหาค่ามากที่สุด (Maximum) ของค่าในตัวแปรอาร์เรย์จำนวน 100 ค่าคือ prices[100] สามารถกระทำได้ด้วยวิธีการดังนี้คือ ใช้การวนรอบ for และ เงื่อนไข if เพื่อการตรวจสอบซึ่งจะทำให้ได้โปรแกรมดังนี้

```
maximum = price[0];
for(i = 1; i <= 99; ++i){
    if  (price[i] > maximum)
        maximum = price[i];
}
```

การรับและแสดงค่าของอาร์เรย์

(Input and Output of Array Values)

ในการให้ค่าแก่ตัวแปรสามารถทำได้ด้วยการกำหนดให้โดยตรงหรือจะให้ผู้ใช้กรอกโดยการรับค่าจากคีย์บอร์ด โดยใช้ร่วมกับคำสั่ง `scanf()` ตัวอย่างเช่น

```
price[5] = 10.69;
scanf("%d %lf", &grades[0], &price[2])
scanf("%c", &code[0]);
scanf("%d %d %d", &grades[0], &grades[1], &grades[2]);
```

หรืออาจใช้การวนรอบเพื่อรับค่าจากผู้ใช้ก็เป็นได้ดังตัวอย่าง

```
for(i = 0; i <=4; ++i)
{
 printf("Enter a grade: ");
 scanf("%d", &grades[i]);
}
```

จะเป็นการวนรอบรับค่า `grades` จากผู้ใช้จำนวน 5 ค่า

การแสดงค่าของอาร์เรย์ในแต่ละสมาชิกสามารถใช้งานร่วมกับคำสั่งเช่น `printf()` ซึ่งทำงานร่วมกับการวนรอบ `for` จะทำให้สะดวกมากยิ่งขึ้น เช่น

- `printf("%f", price[6]);`

  แสดงค่าของ price ที่สมาชิกที่ 6

- `printf("The value of element %d is %d", i, grades[i]);`

  แสดงค่าของ `i` และ ค่า `grades` ที่ `i`

- ```
  for( n = 5; n <= 20; ++n)
   printf("%d %f", n, price[n]);
  ```

  แสดงค่าของ `price` ที่สมาชิกที่ 5 ถึง 20 โดยการใช้งานการวนรอบ
  ```

การให้ค่าเริ่มต้น (Array Initialization)

การกำหนดค่าเริ่มต้นของตัวแปรแบบอาร์เรย์ 1 มิติ สามารถกำหนดได้สำหรับทุกชนิดข้อมูล ดังตัวอย่างต่อไปนี้

```
int grades[5] = {98,87,92,79,85};
char codes[6] = {'s', 'a', 'm', 'p', 'l', 'e'};
double width[7] = {10.96,6.43,2.58,0.86,5.89,7.56
,8.22};
int temp[4] = {10,20,30,40};
float temp[4] = {98.6, 97.2, 99.0 , 101.5};
```

การเรียงข้อมูลคือค่าแรกจะถูกกำหนดให้กับสมาชิกดัชนีที่ 0 และ ค่าที่สองจะถูกกำหนดให้อีลีเมนท์ดัชนีที่ 1 และ ถัด ๆ ไป

ถ้ามีการกำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์ไม่ครบ ก็จะทำการให้ค่าเริ่มต้นเฉพาะค่าที่มีการกำหนดส่วนที่เหลือจะกำหนดให้เป็นศูนย์ เช่น

```
float length[7] = {7.8, 6.4, 4.9, 11.2};
```

จะได้ว่าสี่ค่าแรกคือ length[0] ถึง length[3] จะมีค่าเริ่มต้น ส่วน length[4] ถึง length[6] จะถูกกำหนดให้มีค่าเป็นศูนย์ ในการกำหนดค่าเริ่มต้นจะไม่สามารถทำการกำหนดค่าเริ่มต้นของ อีลีเมนท์ตัวหลังได้ ในขณะที่ยังไม่มีการกำหนดค่าให้กับอีลีเมนต์ตัวแรก ๆ และถ้าไม่มีการกำหนดค่าเริ่มต้น

ในการประกาศตัวแปรแบบอาร์เรย์พร้อมทั้งกำหนดค่าเริ่มต้น สามารถทำได้โดยไม่จำเป็นต้องกำหนดขนาดของอาร์เรย์ ก็ได้เช่น

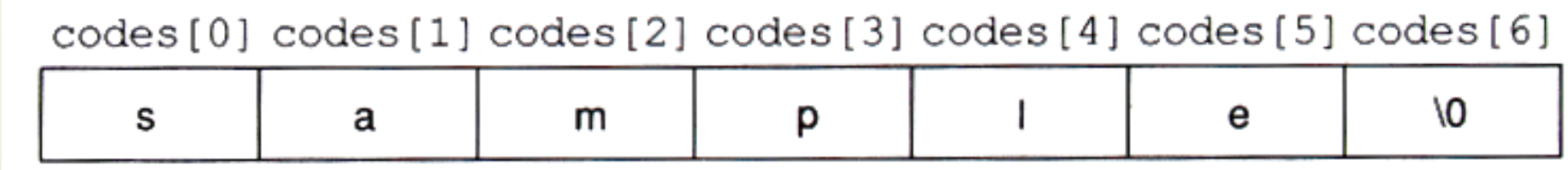
```
char codes[6] = {'s', 'a', 'm', 'p', 'l', 'e'};
char codes[]  = {'s', 'a', 'm', 'p', 'l', 'e'};
```

ทั้งสองบรรทัดทำงานเหมือนกัน โดยตัวแปลภาษา จะทราบด้วยตัวเองว่าขนาดของ อาร์เรย์จะมีขนาดเป็น 6

ในกรณีให้ค่าเริ่มต้นแก่ตัวแปรอักษรสามารถกระทำได้ดังนี้

- char code[] = “sample”;

ซึ่งทำให้เข้าใจได้ว่า สายตัวอักษร หรือเรียกว่า สตริง (String) คือลำดับของตัวอักษรที่เรียงต่อกันนั้นเอง แต่ในกรณีที่เป็นการให้ค่าแบบใช้เครื่องหมายคำพูด หรือแบบสายอักษรนี้ จะทำให้ขนาดของอาร์เรย์มีขนาดเพิ่มจากเดิมมาหนึ่งค่า (ดังรูป) เพื่อใช้ในการเก็บ “\0” หรือเรียกว่า “*null character*” ซึ่งจะถูกเพิ่มเข้าไปโดยอัตโนมัติในส่วนท้ายของสายอักษรซึ่งมีค่าเท่ากับ 0 ทำหน้าที่ในการบอกการสิ้นสุดของสายอักษร



จงเขียนโปรแกรมสำหรับผสานลิสต์ข้อมูลที่จัดเรียงแล้ว

- รับข้อมูลทั้งสองลิสต์
- จัดเรียงข้อมูลทั้งสองลิสต์
- ผสานข้อมูลทั้งสองสิสต์
  - A = {2, 3, 5, 7, 9}
  - B = {1, 4, 6, 8, 10, 11, 12}
  - C = A u B = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

ตัวแปรอาร์เรย์แบบ 2 มิติ

ตัวแปรอาร์เรย์แบบ 2 มิติจะเป็นตัวแปรที่มีการอ้างอิงถึงค่าข้อมูลโดยใช้ค่าเลขดัชนี 2 ค่า ซึ่งประกอบไปด้วยค่าดัชนีที่ใช้ในการอ้างอิงในแนวแถว (row) และค่าดัชนีที่ใช้อ้างอิงในแนวคอลัมน์ (column) ตัวอย่างเช่น

- int val[3][4];

ซึ่งมีข้อมูลคือ

8	16	9	52
3	15	27	6
14	25	2	10

อาร์เรย์แบบนี้เรียกว่าอาร์เรย์ 2 มิติของจำนวนเต็ม ซึ่งประกอบด้วย 3 แถว 4 คอลัมน์ การประกาศตัวแปรอาร์เรย์ 2 มิติจะคล้ายกับ 1 มิติแต่ต้องเพิ่มเติมการกำหนดขนาด ซึ่งจะต้องระบุทั้งในแนวแถวและคอลัมน์

การกำหนดค่าเริ่มต้นในตัวแปรอาร์เรย์ 2 มิติสามารถกระทำได้เช่นเดียวกันกับตัวแปรอาร์เรย์ 1 มิติโดยจะเพิ่มส่วนการแบ่งแยกในแต่ละแถวด้วยเครื่องหมาย “{ }” และ “,” เช่น

```
int val[3][4] = {{8, 16, 9, 52},
                 {3, 15, 27, 6},
                 {14, 25, 2, 10}};
```

การให้ค่าเริ่มต้นแก่ตัวแปรอาร์เรย์ 2 มิติยังสามารถที่จะละเครื่องหมายปีกกาในแต่ละแถวได้โดยให้ใช้เครื่องหมายจุลภาคแทนเช่น

```
int val[3][4] ={8,16,9,52,
                3,15,27,6,
                14,25,2,10};
```

หรือสามารถเขียนได้เป็น

```
int val[3][4] = {8,16,9,52,3,15,27,6,14,25,2,10};
```

```
int main()
{
    int val[3][4] ={8,16,9,52,
                    3,15,27,6,
                   14,25,2,10};
    int i,j;

    for (i=0; i<3; i++) {
        for (j=0; j<4; j++) {
            printf("%3d,", val[i][j]);
        }
        printf("\n");
    }
    return(0);
}
```

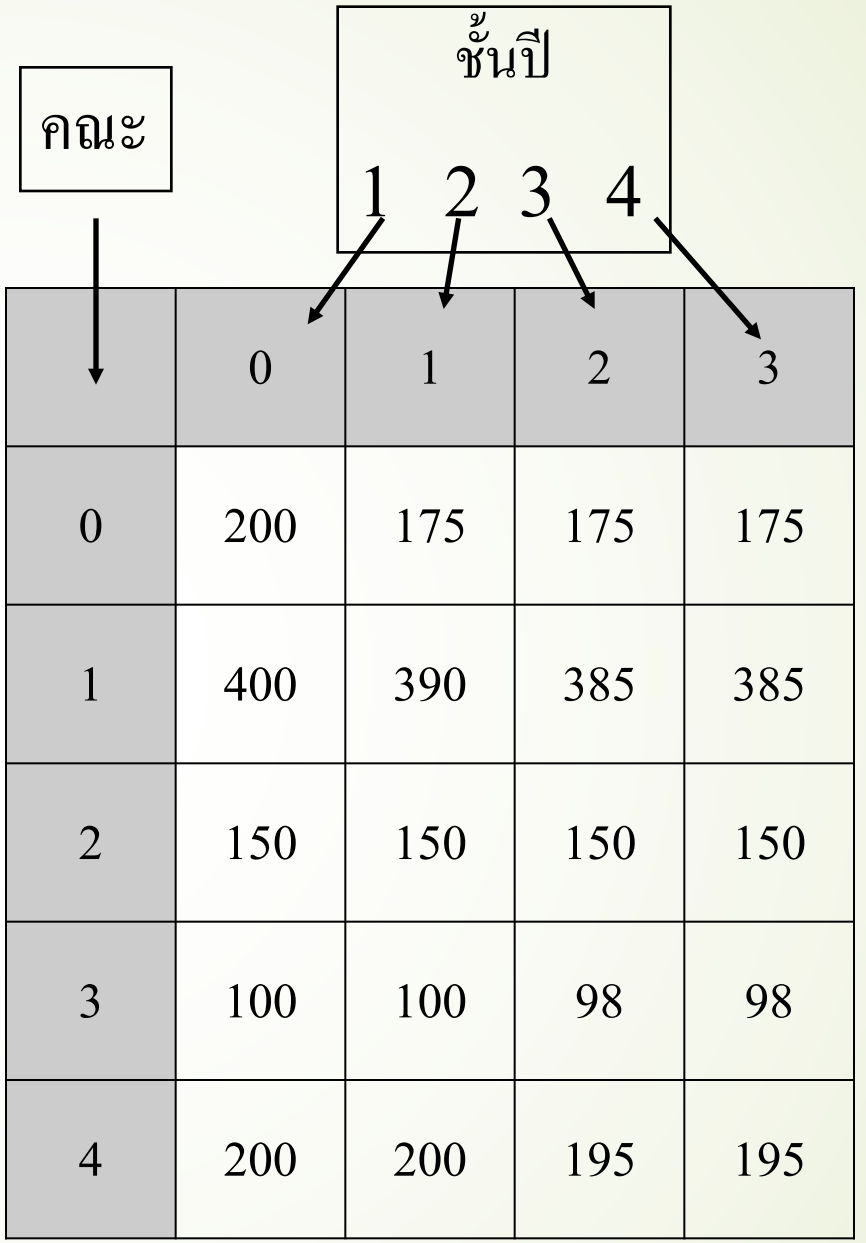
ตัวอย่าง

จงแสดงข้อมูลในตารางดังกล่าวโดยใช้อาร์เรย์

ชั้นปี / คณะ	1	2	3	4
วิทยาศาสตร์	200	175	175	175
วิศวกรรมศาสตร์	400	390	385	385
อักษรศาสตร์	150	150	150	150
รัฐศาสตร์	100	100	98	98
บัญชี	200	200	195	195

```
#include <stdio.h>
void main()
{
   char student[4];
   int num_student[5][4],i,j;
   char fac[20];
   for(i=0 ; i<=4 ; ++i){
      printf("Faculty : ");
      scanf(“%s”, fac);
      for(j=0 ; j<=3 ; ++j){
         printf("student[%d][%d] = ",i,j);
         scanf(”%d”, &num_student[i][j]);
      }
   }
}
```

จากตัวอย่างเป็นการป้อนจำนวนนักศึกษาในตาราง ไปเก็บในอาร์เรย์ num_student ซึ่งมี ขนาด 5x4 = 20 อาร์เรย์สตริงก์ fac ใช้เก็บชื่อคณะต่างๆ การป้อนข้อมูลเริ่มจากจำนวนนักศึกษาของคณะวิทยาศาสตร์ (Row = 0) ชั้นปีที่ 1, 2, 3 และ 4 แล้วจึงตามด้วยจำนวนนักศึกษาของคณะวิศวกรรมศาสตร์จนกระทั่งถึง จำนวนนักศึกษาชั้นปีที่ 4 ของคณะบัญชี อาร์เรย์ num_student จะมีข้อมูลดังนี้



	0	1	2	3
0	200	175	175	175
1	400	390	385	385
2	150	150	150	150
3	100	100	98	98
4	200	200	195	195

ความผิดพลาดของโปรแกรมทั่วไปเกี่ยวกับอาร์เรย์

- ลืมประกาศตัวแปร
- ใช้ตัวแปรนอกขอบเขตที่ประกาศไว้ กรณีนี้คอมไพล์เลอร์จะไม่แจ้งข้อผิดพลาดให้ แต่จะทำให้โปรแกรมผิดพลาดเมื่อใช้ for ในการให้ค่าแก่อาร์เรย์แต่กำหนดจำนวนรอบผิดเนื่องจากให้ค่าเงื่อนไขของ for ไม่ครบทุกสมาชิก
- ลืมให้ค่าเริ่มต้น แม้ว่าส่วนใหญ่คอมไพล์เลอร์จะให้ค่าเริ่มต้นอาร์เรย์ของจำนวนเต็มเป็นศูนย์และตัวแปรแบบอักษรจะเป็นช่องว่าง แต่บางครั้งก็ไม่แน่นอน ดังนี้เพื่อความถูกต้องควรกำหนดค่าเริ่มต้อนก่อนการใช้งาน

แบบฝึกหัด

- จงเขียนโปรแกรมเพื่อบวกเมตริกซ์ A และ B แล้วเก็บไว้ที่ C ให้เขียนเป็นฟังก์ชัน
- จงเขียนโปรแกรมเพื่อลบเมตริกซ์ A และ B แล้วเก็บไว้ที่ C ให้เขียนเป็นฟังก์ชัน
- จงเขียนโปรแกรมเพื่อคูณเมตริกซ์ A และ B แล้วเก็บไว้ที่ C ให้เขียนเป็นฟังก์ชัน