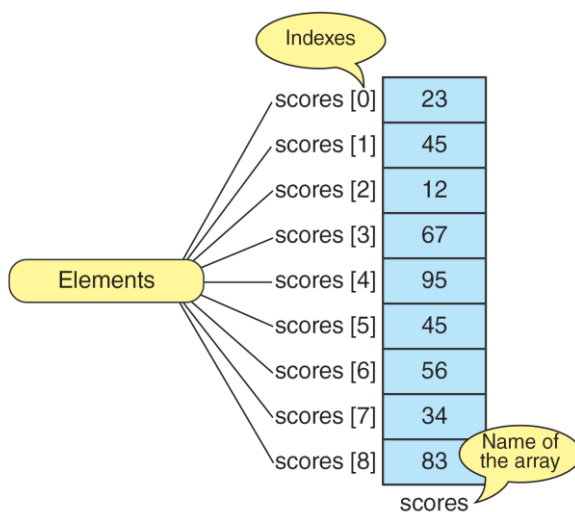




อาร์เรย์คืออะไร

- ตัวแปรประเภทอาร์เรย์ เป็นตัวแปรที่สามารถเก็บข้อมูลหลาย ๆ ค่าไว้ในตัวแปรชื่อเดียวกันได้ โดยระบบจะใช้พื้นที่หน่วยความจำต่อเนื่องกัน เพื่อเก็บข้อมูลชนิดเดียวกันหลายจำนวน ซึ่งจำนวนค่าที่เก็บนั้นขึ้นอยู่กับขนาดของอาร์เรย์ที่ได้กำหนดไว้
- ตัวแปรอาร์เรย์ 1 ตัว สามารถเก็บข้อมูลเป็นชนิดใดก็ได้ แต่ต้องเก็บข้อมูลเป็นชนิดเดียวกันทั้งหมด เช่น
 - ตัวอักษร(char)
 - ชนิดข้อมูลแบบเลขจำนวนเต็ม(int)
 - ชนิดข้อมูลแบบเลขจำนวนจริง(float)

ตัวอย่างของอาร์เรย์



การประกาศตัวแปรอาร์เรย์

การประกาศตัวแปรอาร์เรย์มีรูปแบบดังนี้

ชนิดข้อมูล ชื่อตัวแปร[ขนาด];

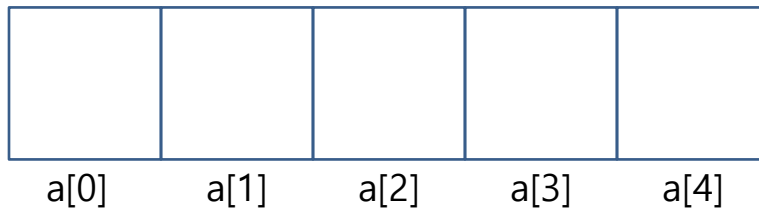
โดย

- ชนิดข้อมูล คือ ชนิดของตัวแปร เช่น int char float
- ชื่อตัวแปร คือ ชื่อของตัวแปร array
- ขนาด คือ ขนาดของตัวแปร

การประกาศตัวแปรอาร์เรย์

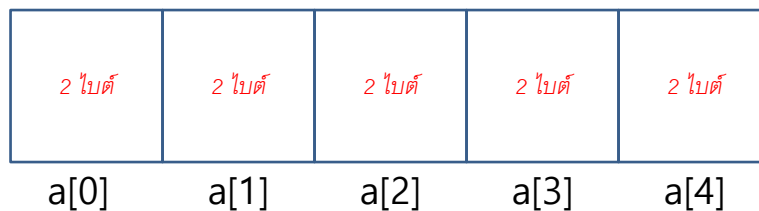
ตัวอย่าง : ประกาศตัวแปรอาร์เรย์เป็น `int a[5];`

จะได้ตัวแปรอาร์เรย์ชื่อ `a` สามารถเก็บข้อมูลเป็นตัวเลขจำนวนเต็ม (integer) ได้ทั้งหมด 5 ตัว ใน `a[0]` `a[1]` `a[2]` `a[3]` `a[4]` โดยมีการจองพื้นที่ในหน่วยความจำลักษณะดังรูป



การประกาศตัวแปรอาร์เรย์

- ปกติข้อมูลประเภทจำนวนเต็ม (integer) 1 ตัวจะใช้พื้นที่หน่วยความจำ 2 ไบต์ ดังนั้นเนื้อที่หน่วยความจำที่ตัวแปรอาร์เรย์ใช้ทั้งหมดจึงเท่ากับจำนวนสมาชิก คูณ ด้วย 2 ไบต์



**ตัวแปรอาร์เรย์ `int a[5];` ใช้หน่วยความจำทั้งหมด 10 ไบต์*

การกำหนดค่าตัวแปรอาร์เรย์

- การกำหนดค่าให้กับตัวแปรอาร์เรย์สามารถกำหนดพร้อมกับการประกาศ เช่น `int a[5]={4,7,2,3,5};` ค่าของตัวแปรจะอยู่ในวงเล็บ {} คั่นด้วยเครื่องหมาย , มีจำนวนเท่ากับขนาดของตัวแปรอาร์เรย์ที่ประกาศไว้ ตัวเลขจะถูกนำไปบรรจุในหน่วยความจำของช่องอาร์เรย์ตามลำดับ

4	7	2	3	5
a[0]	a[1]	a[2]	a[3]	a[4]

การกำหนดค่าตัวแปรอาร์เรย์

- การกำหนดค่าให้กับตัวแปรอาร์เรย์สามารถกำหนดหลังจากการประกาศตัวแปรก็ได้ โดยให้ระบุว่าจะเก็บค่าไว้ในตัวแปรอาร์เรย์ช่องใด เช่น

```
int a[5];
a[0] = 6;
a[3] = 2;
```

6			2	
a[0]	a[1]	a[2]	a[3]	a[4]

การกำหนดค่าตัวแปรอาร์เรย์

- เราสามารถกำหนดตัวแปรอาร์เรย์โดยไม่ระบุขนาดก็ได้ แต่ต้องกำหนดค่าให้กับตัวแปรในวงเล็บ {} โปรแกรมจะทำการกำหนดขนาดของอาร์เรย์ให้อัตโนมัติตามจำนวนสมาชิกที่กำหนด เช่น

```
int a[] = {6,2,4};
```

6	2	4
a[0]	a[1]	a[2]

ตัวอย่างการใช้ตัวแปรอาร์เรย์

- โปรแกรมรับค่าข้อมูลตัวเลขจำนวนเต็ม 10 ตัว แล้วนำมาหาผลรวม

```
#include <stdio.h>
main()
{
    int num[10],sum=0,i;
    for(i = 0; i < 10; i++)
        scanf("%d",&num[i]);
    for(i=0;i<10;i++)
        sum = sum + num[i];
    printf("sum is %d\n",sum);
}
```

ตัวอย่างการใช้ตัวแปรอาร์เรย์

- Histogram – แสดงผลตัวเลขในลักษณะกราฟ โดยแสดงจำนวน * ตามค่าของตัวเลขที่กำหนด

```
#include <stdio.h>
void main() {
    int a[5]={6,3,5,8,4},i,j;
    for(i=0;i<=4;i++){
        printf("%d\t",a[i]);
        for(j=1;j<=a[i];j++){
            printf("*");
        }
        printf("\n");
    }
}
```

```
6    *****
3    ***
5    *****
8    *********
4    ****
```

ตัวอย่างการใช้ตัวแปรอาร์เรย์

- ให้อ่านค่าของจำนวนเต็ม 5 จำนวนจากคีย์บอร์ด และแสดงผลในลำดับที่กลับกัน

```
# include <stdio.h>
main ( ) {
    int k, SIZE = 5;
    int table[SIZE];
    for (k = 0; k < SIZE; k++)
        scanf ("%d", &table[k]);
    for (k = SIZE-1; k >= 0; k--)
        printf ("%d\n", table[k]);
}
```

แบบฝึกหัด

- เขียนโปรแกรมเก็บข้อมูล n ตัว แล้วนำมาหาค่าเฉลี่ย
- เขียนโปรแกรมรับค่าเซตของตัวเลขจำนวนเต็มสองเซต A และ B ซึ่งมีจำนวนสมาชิก 4 ตัว แล้วตรวจสอบว่าทั้งสองเซตมีสมาชิกเหมือนกันกี่ตัว

ตัวแปรอาร์เรย์ 2 มิติ

- ตัวแปรอาร์เรย์ 2 มิติ จะมีลักษณะการเก็บข้อมูลแบบตาราง โดยมีการใช้ index สองตัวในการกำหนดแถว(row) และ คอลัมน์ (column) มีรูปแบบการประกาศตัวแปรดังนี้

ชนิดข้อมูล ชื่อตัวแปร[ขนาดของแถว][ขนาดของคอลัมน์];

เช่น `int a[2][3];` จะได้ตัวแปรอาร์เรย์ที่สามารถเก็บข้อมูลเป็นตัวเลขจำนวนเต็ม(integer) ที่มีจำนวนสมาชิกได้ทั้งหมด $2 \times 3 = 6$ ตัว

การกำหนดค่าตัวแปรอาร์เรย์ 2 มิติ

- การกำหนดค่าให้กับตัวแปรอาร์เรย์ 2 มิติ สามารถกำหนดโดยใช้วงเล็บ {} ได้ดังนี้

```
int a[3][3] = {
    1, 2, 3,
    4, 5, 6,
    7, 8, 9
};
```

```
int b[2][2] = { {1,2},{3,4} };
```

การกำหนดค่าตัวแปรอาร์เรย์ 2 มิติ

```
int a[3][3] = {
    1, 2, 3,
    4, 5, 6,
    7, 8, 9
};
```

หรือ

```
int a[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

การกำหนดค่าตัวแปรอาร์เรย์ 2 มิติ

```
int a[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

=

```
int a[3][3];
a[0][0] = 1; a[0][1] = 2; a[0][2] = 3;
a[1][0] = 4; a[1][1] = 5; a[1][2] = 6;
a[2][0] = 7; a[2][1] = 8; a[2][2] = 9;
```

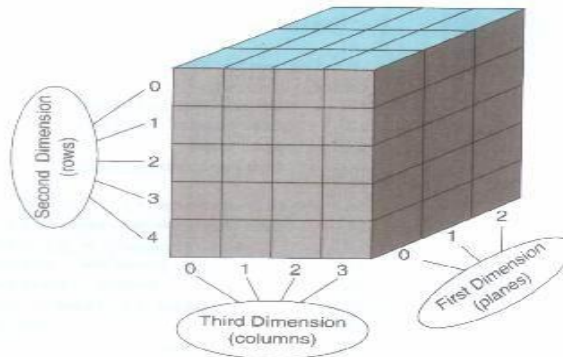
ตัวอย่างตัวแปรอาร์เรย์ 2 มิติ

- โปรแกรมรับค่าตัวเลขจำนวนเต็มจากตัวแปรอาร์เรย์ 2 มิติ แล้วนำข้อมูลทั้งหมดมาบวกกัน

```
int i, j, sum=0, b[5][4];
for(i = 0; i < 5; i++)
    for(j = 0; j < 4; j++)
    {
        scanf("%d", &b[i][j]);
        sum = sum + b[i][j];
    }
printf("The sum is %d\n", sum);
```

ตัวแปรอาร์เรย์ 3 มิติ

- ตัวแปรอาร์เรย์ 3 มิติ อาจเปรียบเทียบกับเป็น arrays of arrays ดังรูป



ตัวอย่างตัวแปรอาร์เรย์ 3 มิติ

- โปรแกรมรับค่าตัวเลขจำนวนเต็มจากตัวแปรอาร์เรย์ 3 มิติ แล้วนำข้อมูลทั้งหมดมาบวกกัน

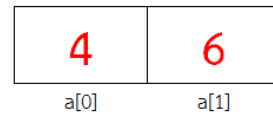
```
int a[3][4][5];
int i, j, k, sum = 0;
for(i = 0; i < 3; i++)
    for(j = 0; j < 4; j++)
        for(k = 0; k < 5; k++)
        {
            scanf("%d", &a[i][j][k]);
            sum = sum + a[i][j][k];
        }
printf("sum is %d", sum);
```

การสลับค่าข้อมูล

- ในการเขียนโปรแกรมจัดการกับค่าหลายค่า ในบางครั้งอาจจะต้องมีการสลับข้อมูลระหว่างตัวแปรสองตัวแปร หรือระหว่างช่องของอาร์เรย์ ซึ่งโดยปกติแล้วการกำหนดค่าให้กับตัวแปรสามารถทำได้โดยใช้เครื่องหมาย = เช่น ต้องการสลับค่าระหว่าง $a[0]$ และ $a[1]$

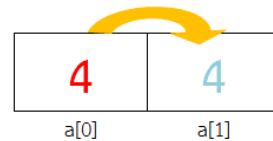
กำหนดค่าเริ่มต้น

`int a[2] = {4,6};`



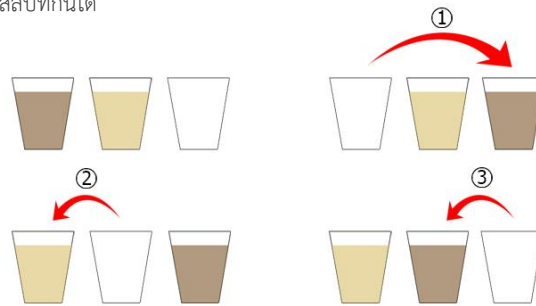
สลับค่า $a[0]$ ไป $a[1]$

`a[1] = a[0];`



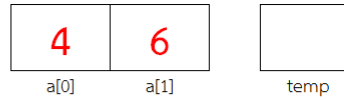
การสลับค่าข้อมูล

- เมื่อทำการสลับค่าในช่องที่ 0 ไป ช่องที่ 1 โดยการเขียนคำสั่ง `a[1]=a[0];` ทำให้ค่าจากช่องที่ 0 ไปอยู่ช่องที่ 1 จริง แต่ก็เกิดปัญหาคือทำให้ค่าเดิมในช่องที่ 1 หายไป เพราะฉะนั้นจึงต้องมีวิธีการสลับค่าข้อมูล โดยให้นักเรียนศึกษาจากกรณีนี้ มีแก้วน้ำอยู่ 2 ใบ ใบหนึ่งใส่กาแฟ อีกใบหนึ่งใส่น้ำ ต้องการสลับน้ำสองแก้ว เราสามารถทำได้โดยเตรียมแก้วเปล่าอีก 1 ใบ มาช่วยในการถ่ายโอน โดยนำกาแฟไปเทใส่แก้วเปล่าก่อน เมื่อแก้วกาแฟว่างก็เทน้ำใส่น้ำแก้วกาแฟ สุดท้ายก็เทกาแฟที่ฝากไว้ในแก้วเปล่าใส่น้ำ ขาก็จะสามารถสลับที่กันได้

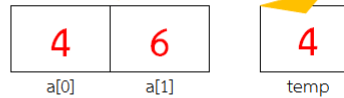


การสลับค่าข้อมูล

`int a[2] = {4,6},temp;`



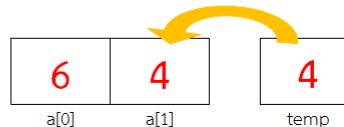
`temp = a[0];`



`a[0] = a[1];`



`a[1] = temp;`



Sorting (การเรียงลำดับข้อมูล)

- Sorting หรือ การเรียงลำดับ คือการจัดเรียงข้อมูลให้อยู่ในรูปแบบ (format) ใดรูปแบบหนึ่ง โดยทั่วไปจะจัดเรียงตามลำดับต่อไปนี้
 - Numerical order (ลำดับตัวเลข)
 - Lexicographical order (ลำดับตัวอักษรตามพจนานุกรม)
- การเรียงลำดับมีความสำคัญเนื่องจากการเรียงลำดับทำให้การค้นหา (search) ข้อมูลมีประสิทธิภาพมากขึ้น ทั้งนี้รวมถึงเร็วขึ้น ทำให้การแก้ปัญหาที่ต้องใช้การเรียงลำดับ หรือ การค้นหาเป็นไปได้เร็วขึ้น

Sorting (การเรียงลำดับข้อมูล)

- ขั้นตอนวิธีที่ใช้ในการจัดเรียงลำดับมีหลายวิธี ดังนี้

1. Bubble sort
2. Insertion sort
3. Merge sort
4. Quick sort
5. Heap sort

*ซึ่งในคลาสเรียนนี้จะศึกษาเฉพาะ *Bubble sort* ซึ่งเป็นการจัดเรียงที่ทำความเข้าใจง่ายและนิยมใช้มากที่สุด

Bubble sort

- เป็นขั้นตอนวิธีการเรียงลำดับที่จะทำการเปรียบเทียบสมาชิกที่อยู่ติดกัน เมื่อพบตำแหน่งที่ผิด (นั่นคือตัวหน้ามากกว่าตัวหลังในกรณีการเรียงจากน้อยไปมาก) ก็จะมีการสลับข้อมูลกัน และจะดำเนินการซ้ำแบบนี้ไปเรื่อยๆจนกว่าจะไม่มีตำแหน่งที่ผิดอีกซึ่งบ่งบอกว่ารายการนั้นเรียงแล้ว ชื่อของขั้นตอนวิธีนี้มีมาจากสมาชิกที่น้อยที่สุดจะค่อยๆถูกสลับขึ้นมาจนอยู่หน้าสุดของรายการ เปรียบได้กับฟองที่ค่อยๆผุดขึ้นมาถึงผิวน้ำ

Bubble sort

Bubble Sort: How does it work?



หัดดาว ปานสมบัติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสงขลานครินทร์ ปัตตานี

Bubble sort

Bubble Sort: How does it work?



หลังรอบที่ 2

ค่าที่มากที่สุดเป็นลำดับ 2 คือ 34 จะไปอยู่ที่ตำแหน่งรองสุดท้ายคือตำแหน่งที่ 3

หัดดาว ปานสมบัติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสงขลานครินทร์ ปัตตานี

Bubble sort

Bubble Sort: How does it work?



หลังรอบที่ 3

ค่าที่มากเป็นลำดับ 3 คือ 28 จะอยู่ที่ตำแหน่งก่อนรองสุดท้าย คือตำแหน่งที่ 2

ทัตดาว ปานสมบัติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสงขลานครินทร์ ปัตตานี

Bubble sort

Bubble Sort: How does it work?



หลังรอบที่ 4

ค่าที่มากเป็นลำดับ 2 คือ 15 จะอยู่ที่ตำแหน่งที่ 1

ข้อสังเกต

ในรอบใดก็ตามที่ไม่มีการสลับค่า แสดงว่าข้อมูลทั้งหมดถูกเรียงลำดับแล้ว

ทัตดาว ปานสมบัติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสงขลานครินทร์ ปัตตานี

Bubble sort

```
int a[5]={15,34,28,36,11},i,j,temp;
for(j=3;j>=0;j--){
    for(i=0;i<=j;i++){
        if(a[i]<a[i+1]){
            temp=a[i];
            a[i]=a[i+1];
            a[i+1]=temp;
        }
    }
}
for(i=0;i<=4;i++) printf("%d ",a[i]);
```

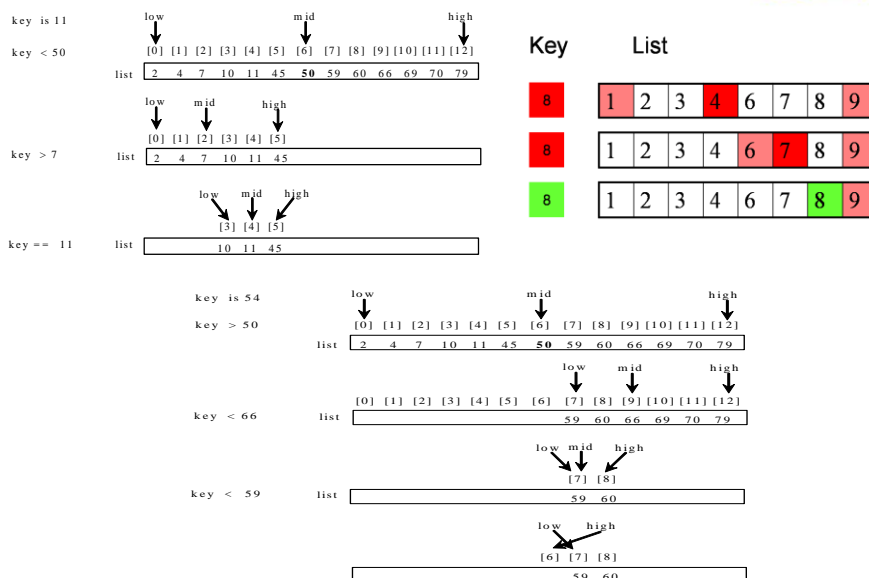
การค้นหาค่าข้อมูลในอาร์เรย์

Key	List
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8

การค้นหาข้อมูลในอาร์เรย์

```
#include <stdio.h>
int main(){
    int key, i, found=0, ansInd;
    int list[] = {1, 4, 9, 3, 5, -3, 6, 2};
    printf("Please enter number to be search:");
    scanf("%d", &key);
    for(i = 0; i < 8; i++){
        if (key == list[i]){
            ansInd=i;
            found=1;
            break;
        }
        else{
        }
    }
    if(found){
        printf("%d is in position %d", key, ansInd+1);
    }
    else{
        printf("Not found");
    }
    return 0;
}
```

Binary Search



Binary Search

```
#include <stdio.h>
int main(){
    int key, i, found=0,ansInd;   int low = 0;  int high = 7;  int mid;
    int unsortedlist[] = {1, 4, 9, 3, 5, -3, 6, 2};
    int list[] = {-3,1, 2, 3, 4, 5, 6, 9 };
    printf("Please enter number to be search:");
    scanf("%d",&key);
    while(low<=high){
        mid = (low + high) / 2;
        printf("mid=%d",mid);
        if (key == list[mid]){
            ansInd= mid;
            found=1;
            break;
        }
        else if (key < list[mid]){
            high = mid - 1;
        }
        else{
            low = mid + 1;
        }
    }

    if(found){
        printf("%d is in position %d",key,ansInd+1);
    }
    else{
        printf("Not found");
    }
    return 0;
}
```

แหล่งข้อมูล

- โครงสร้างข้อมูลแบบอาร์เรย์ (Arrays) , อ.สถิต ประสมพันธ์ ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
- Sorting , อ.พัทธาว ปานสมบัติ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี
- ข้อมูลชนิดอาร์เรย์,สตริง,พอยเตอร์ , เอกสารประกอบการอบรม สอวน.สาขาคอมพิวเตอร์ ศูนย์โรงเรียนสามเสนวิทยาลัย
- ภาษาซีฉบับภาษาชาวบ้าน , กวินวิษณุ พุ่มสาขา ศูนย์เทคโนโลยีเพื่อการเรียนการสอน โรงเรียนสตรีอ่างทอง