

# CS102 Array

Natsuda Kaothanthong  
(19 October, 2015)

## This Lecture

- Array
  - Declaration การประกาศอะเรย์ (Array)
  - การดึงค่าแต่ค่าที่อยู่ในอะเรย์
  - การเรียกฟังก์ชันโดยใช้อะเรย์
  - อะเรย์ของตัวอักษร (String)

## Why do we need an array?

- จงเขียนโปรแกรมที่รับเลยจำนวนเต็ม 5 จำนวน

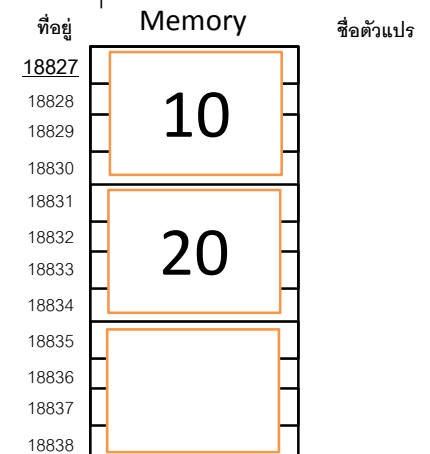
```
int num1, num2, num3, num4, num5;  
printf("Enter 1st number:");  
scanf("%d",&num1);  
printf("Enter 2nd number:");  
scanf("%d",&num2);  
printf("Enter 3rd number:");  
scanf("%d",&num3);  
printf("Enter 4th number:");  
scanf("%d",&num4);  
printf("Enter 5th number:");  
scanf("%d",&num5);
```

หากมีจำนวนตัวเลขที่ต้องการ  
100,000 จำนวน ?

## Array

- ตัวแปรของกลุ่มข้อมูล
- ชื่อของตัวแปร 1 ชื่อ ใช้แทนข้อมูลหลาย ๆ ตัว

```
int num[2] = {10,20};
```



## การประกาศตัวแปร array

ต้องมีการกำหนดขนาดของอะเรย์ตอนประกาศตัวแปร

```
int a[] = {10, 20, 30, 40, 50};
```

```
int a[5] = {10, 20, 30, 40, 50};
```

```
int a[3];
```

```
a[0] = 10; a[1] = 20; a[2] = 30;
```

```
int a[]; 
```

## การเข้าถึงค่าแต่ละค่าที่เก็บใน Array

- ใช้ index
  - index เริ่มต้น 0
  - index สุดท้าย ขนาดของArray – 1
- ตัวอย่าง
- ```
int a[] = {10, 20, 30, 40, 50};
```

  - ```
printf("%d", a[2]);
```
  - ```
int x = a[3]/a[0];
```

`a[index]` เปรียบเสมือนตัวแปร primitive type ตัวหนึ่ง

## การใช้ชื่อตัวแปร[index]

- ```
int num[4] = {8, 12, 4, 7};
```

- ```
num[0] = 20;
```

- ```
scanf("%d", &num[1]);
```

- ```
num[2] = num[0]*2;
```

- ```
num[3] = num[1];
```

- ```
printf("Value of num[4] is %d", num[4]);
```

| ที่อยู่ | Memory | ชื่อตัวแปร |
|---------|--------|------------|
| 18819   |        |            |
| 18823   |        |            |
| 18827   |        |            |
| 18831   |        |            |
| 18835   |        |            |

## ข้อควรระวังในการใช้ index

1. ตำแหน่งของข้อมูลตัวแรกสุดใน Array คือ 0
2. ตำแหน่งของข้อมูลตัวสุดท้ายใน Array คือ ขนาดของArray – 1
3. ในภาษาซี **ไม่มี**คำสั่งในการบอกขนาดของ Array ยกเว้นString

| ที่อยู่ | Memory | ชื่อตัวแปร               |
|---------|--------|--------------------------|
| 18819   |        | int num[4]={8, 12,4, 7}; |
| 18823   |        |                          |
| 18827   |        |                          |
| 18831   |        |                          |
| 18835   |        |                          |

1. เขียนรูปเพื่อแสดงค่า จากตำแหน่งแรก ไปถึงตำแหน่งสุดท้าย
2. เขียนรูปเพื่อแสดงค่า จากตำแหน่ง สุดท้ายไปถึงตำแหน่งแรก

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

```
#define SIZE 5
```

```
void main()
```

```
{
```

```
    int i;
```

```
    int a[SIZE];
```

```
    for(i = 0; i < SIZE; i++)
```

```
    {
```

```
        printf("Enter an integer:");
```

```
        scanf("%d",&a[i];
```

```
    }
```

```
    printf("Entered number are:");
```

```
    for(i = 0; i < SIZE; i++)
```

```
    {
```

```
        printf("%d ",a[i];
```

```
    }
```

```
}
```

โปรแกรมรับค่าจากผู้ใช้และเก็บแต่ละค่าลงในแต่ละตำแหน่งใน **Array** และพิมพ์แต่ละค่าที่ใส่ภายใน **Array**

1. เขียนโปรแกรมรับค่าจำนวนตัวเลขที่ผู้ใช้ต้องการใส่
2. รับค่าตัวเลขแต่ละตัวที่ผู้ใช้ใส่เข้ามา
3. คำนวณค่าเฉลี่ยของตัวเลขที่รับเข้ามา
4. จงหาจำนวนที่น้อยที่สุดที่ผู้ใช้ใส่เข้ามา

การส่งผ่าน**Array**ให้ฟังก์ชันอื่นโดยใช้ **call-by-reference**

- ทำได้ 3 แบบ

1. ใช้ตัวแปร **pointer**

```
void fn1(int *p, int size)
```

```
{
```

คำสั่ง

```
}
```

อยู่ในหัวข้อ  
**Pointer**

2. ใช้ตัวแปร**Array**ที่กำหนดขนาด

```
void fn2(int p[10], int size)
```

```
{
```

คำสั่ง

```
}
```

2. ใช้ตัวแปร**Array**ที่ไม่กำหนดขนาด

```
void fn2(int p[ ], int size)
```

```
{
```

คำสั่ง

```
}
```

```
double average(int p[], int size)
{
    double average =0.0;
    int i;
    for(i = 0; i<size; average+=p[i], i++);
    return average/size;
}
```

```
void main
{
    int balance[] = {1000,2,3,17,50};
    double average;
    average = average(balance, 5);
}
```

## String

### String: Array ของ ตัวอักษร



- กลุ่มข้อมูลชนิดตัวอักษร
- ข้อมูลตัวสุดท้ายของกลุ่มต้องเป็น `'\0'`

```
char name[6] = "Hello";
```

| ที่อยู่ | Memory | ชื่อตัวแปร |
|---------|--------|------------|
| 18827   |        |            |
| 18828   |        |            |
| 18829   |        |            |
| 18830   |        |            |
| 18831   |        |            |
| 18832   |        |            |
| 18833   |        |            |
| 18834   |        |            |

| index     | 0 | 1 | 2 | 3 | 4 | 5  |
|-----------|---|---|---|---|---|----|
| character | H | e | l | l | o | \0 |

### การประกาศตัวแปรชนิด string

- `char word1[6] = "Hello123";` 
- `char word2[] = "Hello2";`
- `char word3[100]; scanf("%s",string1);`
- `char word4[];` 

## Stringและคำสั่ง scanf()

```
int main()
{
    char string1[100];

    scanf("%s",string1);

    return 0;
}
```

name[index] ใช้ในการเข้าถึงตัวอักษรแต่ละตัว

```
int main()
{
    char name[6] = "Hello";
    int i;
    for(i = 0; i<6; i++)
    {
        printf("%c ",name[i]);
    }
    return 0;
}
```

## การคัดลอกระหว่าง string

```
char[6] name1 = "Anne";
char[6] name2 = "Amy";
name1 = name2;
```



## การใช้Cฟังก์ชันของ string

- #include <string.h>
- ฟังก์ชันสำหรับคัดลอกstring 2 ตัว  
strcpy(target string, string ที่ต้องการคัดลอก);
- ฟังก์ชันสำหรับต่อstring 2 ตัว  
strcat(stringที่1, stringที่2);
- ฟังก์ชันหาขนาดของstring  
ตัวแปรสำหรับเก็บขนาด = strlen(stringที่ต้องการหาขนาด);

## การใช้Cฟังก์ชันของ string

- การเปรียบเทียบstring 2 ตัว ใช้เพื่อเรียงลำดับstring

`strcmp(stringที่1, stringที่2);`

ผลที่ได้มีค่า -1 ถ้า stringที่1 มาก่อน stringที่2

- เช่น `strcmp("abc", "def");`

ผลที่ได้มีค่า 0 ถ้า stringที่1และstringที่2เหมือนกัน

- เช่น `strcmp("abc", "abc");`

ผลที่ได้มีค่า 1 ถ้า stringที่1 มาทีหลัง stringที่2

- เช่น `strcmp("def", "abc");`

เขียนโปรแกรมเพื่อเรียงลำดับชื่อเพื่อ 2 ชื่อที่รับเข้ามาให้อยู่ในลำดับจากa ไปหาz