

Chapter 8 : One-dimensional Arrays



- Array Basic
- Array Declaration and Initialization
- Accessing array elements
- Array and Loop



```
graph TD; CDataTypes[C Data Types] --> FundamentalTypes[Fundamental Types]; CDataTypes --> DerivedTypes[Derived Types]; FundamentalTypes --> Integer[Integer]; FundamentalTypes --> Floating[Floating]; FundamentalTypes --> Character[Character]; Integer --> IntegerExample[int age = 46;]; Floating --> FloatingExample[float temperature = 30.29;]; Character --> CharacterExample[char letter = 'Z']; DerivedTypes --> Array[Array]; DerivedTypes --> Pointer[Pointer]; DerivedTypes --> Structure[Structure]; DerivedTypes --> Union[Union]; Array --> ArrayDescription[เก็บค่าได้เพียง 1 ค่า (Single value)]; Pointer --> PointerDescription[ต่อตัวแปร 1 ตัว เช่น]; Structure --> StructureDescription[เก็บค่าได้มากกว่า 1 ค่า (Multiple value)]; Union --> UnionDescription[ต่อตัวแปร 1 ตัว เช่น];
```

C Data Types

- Fundamental Types**
 - Integer**: int age = 46;
 - Floating**: float temperature = 30.29;
 - Character**: char letter = 'Z';
- Derived Types**
 - Array**: เก็บค่าได้เพียง 1 ค่า (Single value)
 - Pointer**: ต่อตัวแปร 1 ตัว เช่น
 - Structure**: เก็บค่าได้มากกว่า 1 ค่า (Multiple value)
 - Union**: ต่อตัวแปร 1 ตัว เช่น



สมมติให้นิสิตเขียนโปรแกรมเพื่อหาค่าเฉลี่ยของอุณหภูมิในหนึ่งปี (365 วัน) ที่รับมาทางคีย์บอร์ด โดยจะต้องเก็บค่าทุกค่าไว้ในตัวแปร นิสิตจะอย่างไร

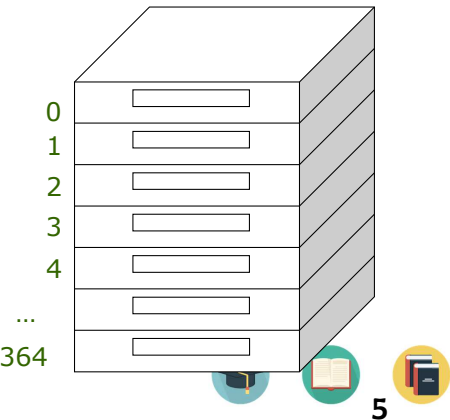
- ```
avg =(temp1+ temp2+temp3+ temp4+...+temp365)/ 365;
```

# 1. Array Basic

- **อาเรย์ (Arrays)** ตัวแปรชุด ที่เก็บข้อมูลประเภทเดียวกัน
- อาเรย์ช่วยอำนวยความสะดวกในการประกาศและใช้งานตัวแปรจำนวนมาก

ตัวอย่าง ประกาศตัวแปร array ชนิดจำนวนจริง 365 ตัว

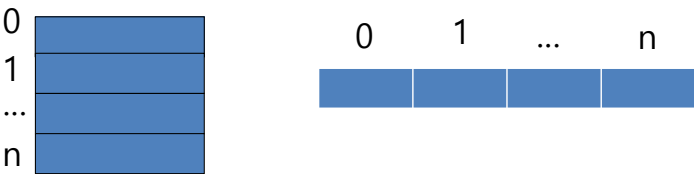
```
float temp[365];
```



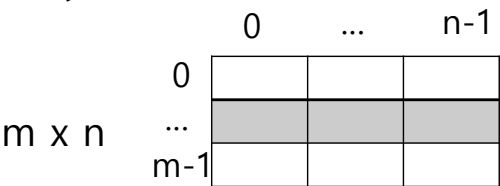
# 1. Array Basic

- อาเรย์อาจเปรียบเทียบกับ vector หรือ matrix ในทางคณิตศาสตร์

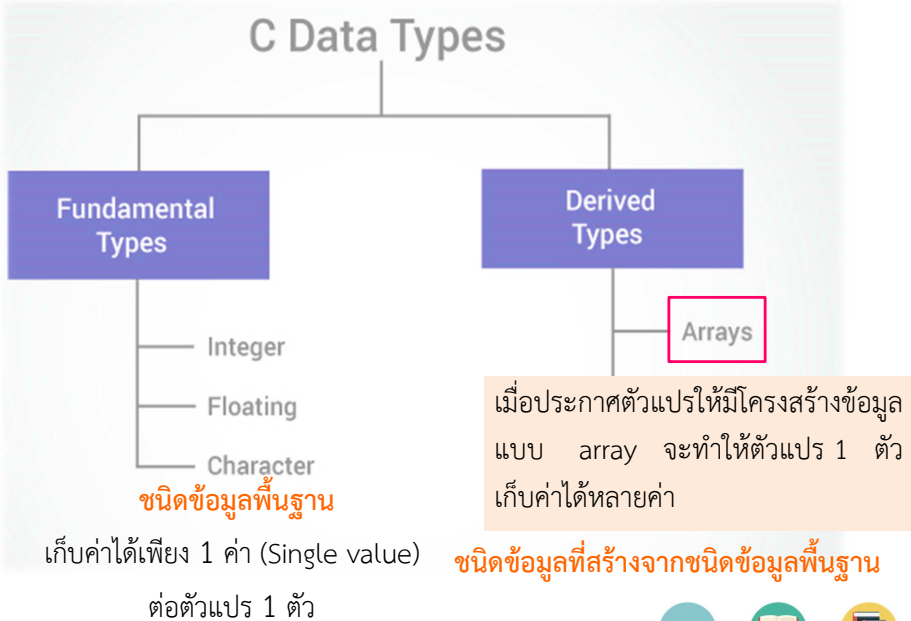
—Array 1 มิติ เปรียบได้กับ vector คือ มีเพียงแถวหรือคอลัมน์



—Array 2 มิติ เปรียบได้กับ matrix ที่มีทั้งแถวและคอลัมน์



## Review: Data types in C



# 1. Array Basic

- ตัวอย่าง โปรแกรมเพื่อหาค่าเฉลี่ยของอุณหภูมิในหนึ่งปี

```
float total=0;
float temp1, temp2, temp3,
temp4,...,temp365;

scanf("%f", &temp1);
...
scanf("%f", &temp365);

total =temp1+temp2+temp3
+temp4+...+temp365;

printf("Avg=%.2f",total/365);
```

```
int i;
float total=0;

float temp[365];

for (i=0;i <365;i++)
scanf("%f",&temp[i]);

for (i=0;i <365;i++)
total +=temp[i];

printf("Avg=%.2f",total/365);
```



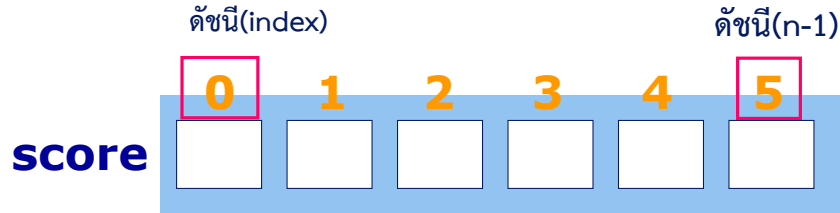
## 2. Array Declaration and Initialization

- รูปแบบ(Syntax) การประกาศตัวแปร ชนิดอาเรย์:

ชนิดข้อมูล ชื่อตัวแปร[จำนวนสมาชิก];

int char float double bool

- ตัวอย่าง: `int score[6];`



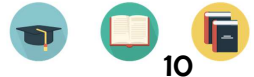
9

## 2. Array Declaration and Initialization

- รูปแบบ(Syntax) การประกาศตัวแปรชนิดอาเรย์ พร้อมกำหนดค่าเริ่มต้น:

ชนิดข้อมูล ชื่อตัวแปร[จำนวนสมาชิก]={ ค่าที่1,ค่าที่2, ...};

- ตัวอย่าง: `int score[4]= {5,6,7,9};`



10

## 2. Array Declaration and Initialization

- ตัวอย่างเพิ่มเติม: การประกาศตัวแปรชนิดอาเรย์

```
int number1[3] = {11, 55, 15};
```



```
int number2[] = {12, 5, 13};
```



หากไม่มีการกำหนดจำนวนสมาชิก  
compiler จะใช้จำนวนค่าเริ่มต้น เป็นจำนวนสมาชิก

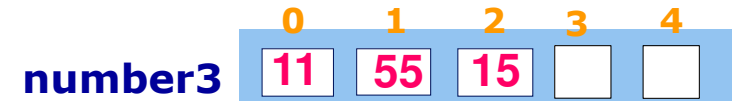


11

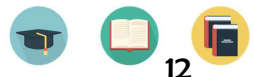
## 2. Array Declaration and Initialization

- ตัวอย่างเพิ่มเติม: การประกาศตัวแปรชนิดอาเรย์

```
int number3[5] = {11, 55, 15};
```



```
int number4[2] = {11, 55, 15};
```

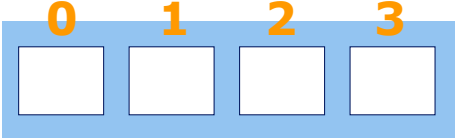


12

### 3. Accessing array elements

- สมาชิกของอาร์เรย์ เรียกว่า **Elements**
- ตัวเลขที่ใช้ระบุสมาชิกของอาร์เรย์ เรียกว่า **ดัชนี (Indexes)**
  - การเข้าถึงสมาชิกแต่ละตัวของอาร์เรย์ จะต้องระบุ ดัชนี
  - ดัชนีของอาร์เรย์ จะเริ่มต้นที่ค่า 0 เสมอ**

```
int score[4];
score[0] = -3; score
score[2+1] = 7;
score[3-1] = score[0]+score[3]
printf("%d",score[2]);
```



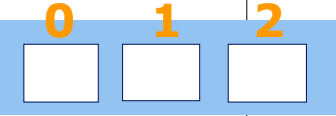
13

### 3. Accessing array elements

- ตัวอย่าง 1:** บวกจำนวนเต็ม 3 ตัวที่รับค่าจากคีย์บอร์ด

```
int nums[____];
scanf("%d",&nums[0]); nums

int sum = nums[0] + _____ + _____ ;
printf("sum=%d",sum);
```



ชื่อตัวแปรอาร์เรย์[index] เทียบเท่ากับ ตัวแปรเดียว 1 ตัว



14

### 3. Accessing array elements

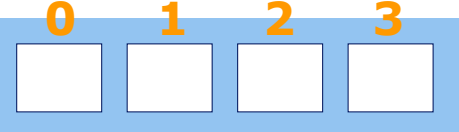
- ข้อควรระวัง:** การระบุดัชนี เกินขอบเขตของอาร์เรย์

```
const int SIZE = 4;
int numbers[SIZE];

numbers[10] = 999;
printf("%d\n", numbers[77]);
```

ดัชนี 10 และ 77 ไม่อยู่ในขอบเขตของอาร์เรย์ (Index out of bound) ทำให้เกิด Runtime Error

**numbers**



จำนวนสมาชิก: 4  
ขอบเขต: 0-3



15

### Quick check1

- จงวาดโครงสร้างของอาร์เรย์ A,B พร้อมทั้งเขียนค่าที่จัดเก็บในสมาชิกแต่ละตัว เมื่อประกาศตัวแปรอาร์เรย์ A, B และกำหนดค่าดังนี้

```
float A[3];

int B[6] = {9};

A[0] = B[0];
A[1] = A[0] *2;
A[2] = A[0]/2;

B[B[0]/3] = 13;
B[5] = B[5-1] + B[5-2]-A[1];
```



16

## 4. Array and Loop

- การเข้าถึงสมาชิกอาเรย์ทุกตัว จะใช้คำสั่งลูป โดยกำหนดให้ตัวนับของลูป เป็นดัชนีของอาเรย์ (ชนิดจำนวนเต็มเท่านั้น)
- ตัวอย่าง 2: บวกจำนวนเต็ม 3 ตัวที่รับค่าจากคีย์บอร์ด (โดยใช้ลูป)

```
int nums[3]
int sum=0,i;
```

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

```
printf("sum=%d",sum);
```

|   |    |             |
|---|----|-------------|
| 0 | 5  | nums [0]=5  |
| 1 | 10 | nums [1]=10 |
| 2 | 20 | nums [2]=20 |

sum=35



17

## 4. Array and Loop

- การเข้าถึงสมาชิกอาเรย์ทุกตัว จะใช้คำสั่งลูป โดยจำนวนรอบของลูป จะเท่ากับ ขนาดของอาเรย์ (array size)
  - ขนาดของอาเรย์ คือ จำนวนสมาชิกของอาเรย์ หาได้โดยการใช้ตัวดำเนินการ sizeof()

```
sizeof(arrayName)/sizeof(arrayName[0]);
```

```
int score[] = {5,6,7,9,10};
int asize;
asize = sizeof(score)/sizeof(score[0]);
printf("array size=%d",asize);
```

size in byte of array=20  
size in byte data type=4  
array size=5



18

## Fun fact

- อาเรย์ในหน่วยความจำ (1D Array in memory)

```
int score[] = {23,4,6,15,5,7};
```

|    |   |   |    |   |   |
|----|---|---|----|---|---|
| 23 | 4 | 6 | 15 | 5 | 7 |
| 0  | 1 | 2 | 3  | 4 | 5 |

Array index

ตำแหน่งในหน่วยความจำ  
(int ขนาด 4 bytes)

5000  
5004  
5008  
5012  
5016  
5020

RAM memory

|    |
|----|
| 23 |
| 4  |
| 6  |
| 15 |
| 5  |
| 7  |

ตำแหน่ง (Address)ในหน่วยความจำของสมาชิกอาเรย์  
แต่ละตัวจะติดต่อกัน



19

## 4. Array and Loop

- ตัวอย่าง 3: กำหนดค่าสมาชิกทุกตัวในอาเรย์ โดยคำสั่งลูป

```
float A[5];
```

```
int i = 0;
```

```
while (i < 5)
```

```
{
```

```
 A[i] = pow(i,2.0);
```

```
 printf("%.0f\n",A[i]);
```

```
 i = i + 1;
```

```
}
```

|  |   |
|--|---|
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |

เขียนโปรแกรมนี้ใหม่ โดยใช้คำสั่ง for



20

## 4. Array and Loop

- ตัวอย่าง 4: การบวก vector  $\vec{u}=(-2,9,3)$

$$\vec{w}=(1,1,5) \quad \vec{x} = \begin{bmatrix} -1 \\ 10 \\ 8 \end{bmatrix}$$

```
int u[]={-2,9,3};
int w[]={1,1,5};
int x[3];
x=u+w; ❌
```

compile error

การดำเนินการทางคณิตศาสตร์  
ไม่สามารถใช้กับตัวแปรอาเรย์ได้โดยตรง  
ต้องแยกคำนวณสมาชิกแต่ละตัวใน  
อาเรย์

```
int i = 0;
for (i=0; i < 3; i++)
{
 x[i] = u[i] + w[i];
 printf("%d\n", x[i]);
}
```



21

## 4. Array and Loop

- ตัวอย่าง 4: การบวก vector  $\vec{u}=(-2,9,3)$

$$\vec{w}=(1,1,5) \quad \vec{x} = \begin{bmatrix} -1 \\ 10 \\ 8 \end{bmatrix}$$

```
int u[]={-2,9,3};
int w[]={1,1,5};
int x[3];
int i = 0;
```

```
for (i=0; i < 3; i++)
{
 x[i] = u[i] + w[i];
 printf("%d\n", x[i]);
}
```

0  
1  
2  
3

End Loop

u[i] + w[i]

-2 + 1 = -1  
9 + 1 = 10  
3 + 5 = 8

-1 0  
10 1  
8 2



22

## Quick check2

- กำหนดให้โปรแกรมมีขั้นตอนการทำงานดังนี้

เริ่มต้นโปรแกรม

รับค่า count

ทำซ้ำ โดยให้ i = 0 ถึง count-1

x[i] = i + 1

จบทำซ้ำ

จบโปรแกรม

ถ้าโปรแกรมนี้อัปเดตผลจนจบ กำหนดให้ count เป็น 7  
ให้หีสิตวาดโครงสร้างของอาเรย์ x และค่าที่เก็บในอาเรย์  
แปลงขั้นตอนการทำงานนี้ เป็นโปรแกรม



## Quick check2

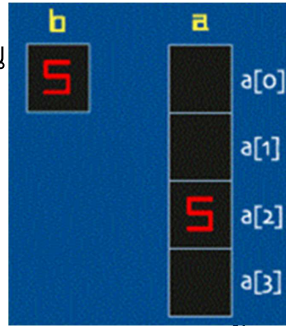
- แปลงขั้นตอนการทำงานนี้ เป็นโปรแกรม



# Summary

- อาร์เรย์ คือ ตัวแปรชุดที่เก็บข้อมูลชนิดเดียวกัน

```
int b=5;
int a[4];
a[2]=5;
```



- การประกาศตัวแปรอาร์เรย์ขึ้นใช้งานนั้น จะต้องกำหนด**ขนาด**ของอาร์เรย์ทุกครั้ง (ยกเว้นการประกาศพร้อมกำหนดค่าเริ่มต้น)

```
int a[10];
int b[]={1,2,3,4,5};
int c[];
```



25

# Summary

- การเข้าถึงสมาชิกแต่ละตัวของอาร์เรย์ จะต้องระบุ ดัชนี (index) ในเครื่องหมาย [ ] ที่อยู่หลังชื่ออาร์เรย์ โดยดัชนีของอาร์เรย์ในภาษา C จะเริ่มต้นที่ 0 เสมอ
- และ อาร์เรย์ในภาษา C ไม่มีการตรวจสอบการอ้างอิงถึงดัชนีที่เกินขอบเขตของอาร์เรย์ (Compile ผ่านแต่ให้ผลการทำงานไม่ถูกต้อง)
- การเข้าถึงสมาชิกอาร์เรย์ทุกตัว นิยมใช้คำสั่งลูปมาช่วย โดยให้ตัวนับของลูป เป็นดัชนีของอาร์เรย์

```
for(i=0;i<4;i++)
 a[i]=5;
```

```
a[0]=5;
a[1]=5;
a[2]=5;
a[3]=5;
```



26