

One-Dimensional Array

w11-Lab

Assembled for 204111
by Ratsameetip Wita

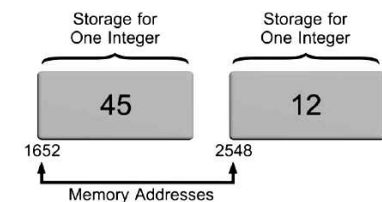
Data Types in C (Revisited)

- ชนิดของข้อมูล และการดำเนินการประเภทใดบ้างที่กระทำกับข้อมูลในแต่ละชนิดได้
- ชนิดของข้อมูลที่เป็นส่วนหนึ่งของภาษา เราเรียกชนิดข้อมูลเหล่านี้ว่า **Primitive Type**
- ตัวอย่าง **Data types** ในภาษา C
 - char, int, short, long, float, double
- ชนิดของข้อมูลจะมีการกำหนดให้กับตัวแปรที่ใช้งานในโปรแกรม

Topics

- **Introduction to Array**
- **1D Array**
 - Declaration
 - Assignment
- **Array of char: String**

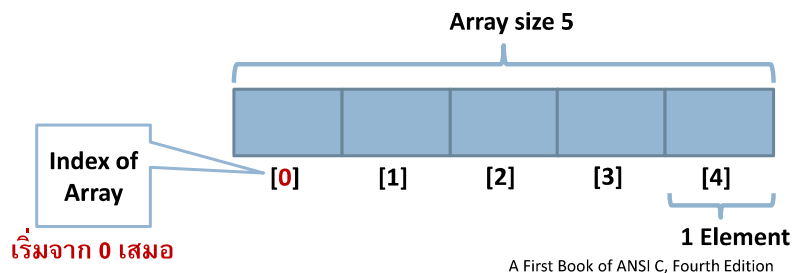
Variables



- ในการใช้งานตัวแปรใด ๆ ใน **declaration** จะมีการจองพื้นที่ในหน่วยความจำเป็นจำนวน **byte** ตามขนาดของ **data type**
- 1 ตัวแปร เก็บค่าข้อมูล 1 ตัว
- หากต้องการสร้างตัวแปรเพื่อเก็บข้อมูลตัวเลขจำนวน 20 ตัว = ประกาศตัวแปร 20 ตัว
 - คะแนนของนักศึกษา 30 คน
 - ความสูงของคน 120 คน

Introduction to Array

- อาร์เรย์ (Array) คือชุดของตัวแปร ที่มี data type ชนิดเดียวกัน
- ในระบบคอมพิวเตอร์จะมีการจองพื้นที่ในหน่วยความจำไว้เป็นชุด
- จำเป็นต้องมีการกำหนดจำนวนข้อมูล (Element) ของชุดข้อมูล (Array) ในขั้นตอนการประกาศตัวแปร (Declaration)



A First Book of ANSI C, Fourth Edition

5

Topics

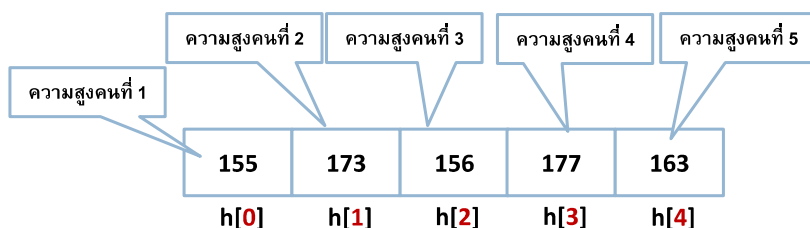
- Introduction to Array
- 1D Array
 - Declaration
 - Assignment
- Array of char: String

A First Book of ANSI C, Fourth Edition

6

One-Dimension Array (1D Array)

- ในการใช้งานตัวแปรแบบอาร์เรย์ จะเป็นการมองข้อมูลประเภทเดียวกันเป็นชุดของข้อมูล (1 ตัวแปรเก็บข้อมูลหลาย element)
- ใช้ index [0] ถึง $[n-1]$ ในการระบุลำดับของข้อมูลในอาร์เรย์ที่มีขนาด n ตัว
 - เช่น ส่วนสูงของคน 5 คน กำหนดให้อาร์เรย์ h เก็บข้อมูลความสูง

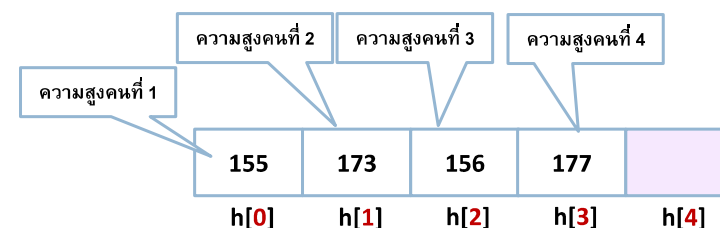


A First Book of ANSI C, Fourth Edition

7

One-Dimension Array (1D Array) (2)

- ในการทำงานกับอาร์เรย์ เนื่องจากในภาษา C ไม่มีกลไกในการตรวจสอบขนาด และจำนวนข้อมูลในอาร์เรย์ ทำให้ต้อง มีการเก็บค่าของจำนวนสมาชิก และขนาดของอาร์เรย์ไว้
- เช่น h เป็นอาร์เรย์มีขนาด 5 และ มีสมาชิก 4



A First Book of ANSI C, Fourth Edition

8

Topics

- Introduction to Array
- 1D Array
 - Declaration
 - Assignment
- Array of char: String

Array Declaration (2)

- อาร์เรย์สามารถประกาศแบบกำหนดค่าเริ่มต้นได้

```
int num[5];
```

```
int num[5] = {4, 2, 5, 7, 6};
```

```
int num[] = {4, 2, 5, 7, 6};
```

← สามารถละการกำหนดขนาดใน [] ได้
กรณีที่มีการกำหนดค่าเริ่มต้นเท่านั้น

ข้อควรระวัง: ไม่สามารถประกาศตัวแปรแบบอาร์เรย์โดยไม่กำหนดขนาดได้

```
int num[]; ❌
```

Array Declaration

- การประกาศตัวแปรชนิดอาร์เรย์ ต้องระบุ data type และขนาดของอาร์เรย์ ทุกครั้ง

```
datatype arrayName[array_size];
```

```
int num[20]; // อาร์เรย์ของข้อมูลแบบ int ที่สามารถเก็บข้อมูลได้ 20 ตัว
```

```
float score[12]; // อาร์เรย์ของข้อมูลแบบ float ที่สามารถเก็บข้อมูลได้ 12 ตัว
```

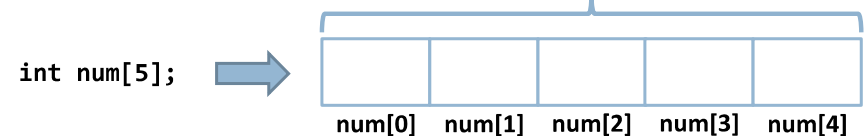
```
#define MAXLENGTH 50
```

```
char name[MAXLENGTH]; // อาร์เรย์ของข้อมูลแบบ char ที่สามารถเก็บข้อมูล  
// ได้ตามขนาด MAXLENGTH
```

Array Declaration (3)

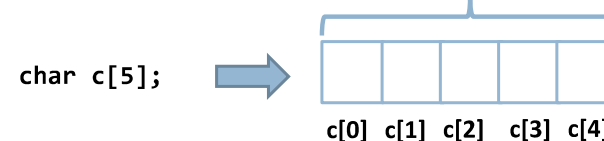
ตัวแปรแบบ **int** จำนวน 5 ตัว

จองพื้นที่ = $5 * 4 \text{ bytes} = 20 \text{ bytes}$



ตัวแปรแบบ **char** จำนวน 5 ตัว

จองพื้นที่ = $5 * 1 \text{ byte} = 5 \text{ bytes}$



Topics

- Introduction to Array
- 1D Array
 - Declaration
 - Assignment
- Array of char: String

Array Assignment (2)

- สามารถใช้ loop เพื่อกำหนดค่า หรืออ่านค่าที่อยู่ในอาเรย์ได้

```
#define NUMELS 5
int grades[NUMELS];
int i;
for(i = 0; i < NUMELS; i++)
{
    printf("Enter a grade: ");
    scanf("%d", &grades[i]);
}
```

Size of Array

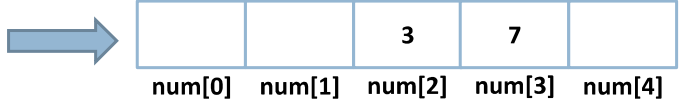
Sample output:

```
Enter a grade: 85
Enter a grade: 90
Enter a grade: 78
Enter a grade: 75
Enter a grade: 92
```

Array Assignment

- ในการกำหนดค่าตัวแปรแบบอาเรย์ จะเป็นการกำหนดค่าให้กับ element ที่อยู่ใน Array
- โดยระบุ element ได้จาก index []

```
num[3] = 7;
num[2] = 3;
```



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

- ข้อควรระวัง: ในภาษา C ไม่มีการตรวจสอบ index ที่เกินขอบเขตที่ประกาศไว้ (boundary check)

```
num[5] = 7; ← Runtime Error!!
```

Input and Output of Array Values

```
01 #include <stdio.h>
02 int main()
03 {
04     #define NUMELS 5
05     int grades[NUMELS];
06     int i;
07     //Input grades
08     for(i = 0; i < NUMELS; i++){
09         printf("Enter a grade: ");
10         scanf("%d", &grades[i]);
11     }
12     //Output index and value of grades[index]
13     for(i = 0; i < NUMELS; i++){
14         printf("Index: %d, Value: %d\n", i, grades[i]);
15     }
16     return 0;
17 }
```

Sample output:

```
Enter a grade: 85
Enter a grade: 90
Enter a grade: 78
Enter a grade: 75
Enter a grade: 92
Index: 0 Value: 85
Index: 1 Value: 90
Index: 2 Value: 78
Index: 3 Value: 75
Index: 4 Value: 92
```

Practice 1: findMax

- ให้เขียนโปรแกรมเพื่อรับค่าจำนวนเต็ม n จำนวน และบันทึกลงใน array หลังจากนั้นให้แสดงตัวเลขที่มีค่ามากที่สุด และลำดับของตัวเลขนั้นในการรับค่า (แสดงเฉพาะ index แรกหากค่ามากที่สุดมีมากกว่า 1 จำนวน)
- ตัวอย่างการ run 1

```
Input n: 6
Input all numbers: 5 2 1 2 9 9
Max number = 9
Index of Max number = 4
```

- ตัวอย่างการ run 2

```
Input n: 10
Input all numbers: 5 23 1 2 9 3 15 7 16 4
Max number = 23
Index of Max number = 1
```

Array of char: String

- ในภาษา c จะมองข้อความ หรือ String อยู่ในรูปแบบของ อาร์เรย์ของตัวอักษร (Array of char)
- เราสามารถกำหนดค่าตัวแปรแบบ array of char ได้ดังนี้

```
char codes[6] = {'s', 'a', 'm', 'p', 'l', 'e'};
char codes[] = {'s', 'a', 'm', 'p', 'l', 'e'};
char codes[] = "sample"; /* size is 7 */
```

s	a	m	p	l	e	\0
---	---	---	---	---	---	----

- C compiler จะต่อท้ายทุก string ด้วย NULL character ('\0') เพื่อบอกจุดสิ้นสุดของ string (array of char)

Topics

- Introduction to Array
- 1D Array
 - Declaration
 - Assignment
- Array of char: String

#include <stdio.h>

String Input and Output

- การอ่าน input string:
 - gets()
 - scanf()
 - getchar()
- การแสดงผล output string
 - puts()
 - printf()
 - putchar()

String Input and Output (2)

- **scanf()** อ่าน characters เข้ามาจนเจอ blank space หรือ newline (บรรทัดใหม่) – (อ่านทีละคำ)
 - `scanf("%s", message);` //ไม่ต้องใส่ &
- **gets()** อ่าน characters เข้ามาจนเจอ newline (บรรทัดใหม่) – (อ่านทีละบรรทัด)
- **printf()** และ **puts()** สามารถใช้แทนกันได้
 - `printf("%s\n", message);` \equiv `puts(message);`

String Formatting

- `char happy[] = "Have a Happy Day"`
- `printf("|%25s|", happy);`
 - | Have a Happy Day |
- `printf("|%-25s|", happy);`
 - |Have a Happy Day |
- `printf("|%25.12s|", happy);`
 - | Have a Happy |
- `printf("|%.12s|", happy);`
 - |Have a Happy|

String Input and Output (3)



Program 9.1

```

1  #include <stdio.h>
2  int main()
3  {
4      #define MSIZE 81
5      char message[MSIZE]; /* enough storage for 80 characters plus '\0' */
6
7      printf("Enter a string:\n");
8      gets(message);
9      printf("The string just entered is:\n");
10     puts(message);
11
12     return 0;
13 }

```

Output

```

Enter a string:
This is a test input of a string of characters.
The string just entered is:
This is a test input of a string of characters.

```

sizeof() function

- ในการทำงานกับตัวแปรแบบอาร์เรย์เราสามารถหาค่าขนาดของอาร์เรย์ (จำนวน **byte** ที่ใช้ทั้งหมด) ได้โดยใช้คำสั่ง

```
int sizeof(Array_Name);
```

- เช่น

Sample output:
size = 7
size = 80

```

char test[] = "sample";
int intArray[20];

printf("size = %d", sizeof(test));
printf("size = %d", sizeof(intArray));

```

s	a	m	p	l	e	\0
---	---	---	---	---	---	----

ไม่ต้องใส่ □

Array of char: Encoding

- ในการเก็บข้อมูล array of char จะมีการเก็บในรูปแบบ ASCII code

s	a	m	p	l	e	\0
---	---	---	---	---	---	----

Sample output:

```
test[0]: ASCII: 115, char: s
test[1]: ASCII: 97, char: a
test[2]: ASCII: 109, char: m
test[3]: ASCII: 112, char: p
test[4]: ASCII: 108, char: l
test[5]: ASCII: 101, char: e
test[6]: ASCII: 0, char:
```

\0

```
int i;
char test[] = "sample";
for(i=0;i<sizeof(test);i++) {
    printf("test[%d]: ASCII: %d, char: %c\n", i, test[i], test[i]);
}
```

Practice 2: Uppercase

- ให้เขียนโปรแกรมรับค่า String (a-z, A-Z) ในรูปแบบ array of char ความยาวไม่เกิน 50 ตัวอักษร จากนั้นให้แปลงตัวอักษรเพื่อแสดงผลในรูปแบบตัวพิมพ์ใหญ่

- ตัวอย่างการ run 1

Input: test
Uppercase: TEST

Character	ASCII (Decimal)
A	65
Z	90
a	97
z	122

- ตัวอย่างการ run 2

Input: ComSci
Uppercase: COMSCI

for ASCII Table: <http://en.wikipedia.org/wiki/ASCII>

Practice 3: Fibonacci Sequence

- ให้รับค่าจำนวนพจน์ n ($1 \leq n \leq 100$) แล้วทำการคำนวณและแสดงผลลำดับฟีโบนัชชี ตามจำนวนพจน์ที่กำหนด โดยลำดับฟีโบนัชชีมีนิยามดังนี้
 - จำนวนในลำดับใด ๆ เท่ากับผลบวกของจำนวนสองจำนวนก่อนหน้านั้น
$$f_n = f_{n-1} + f_{n-2}$$
 - สองจำนวนแรก (พจน์ที่ 1 และ พจน์ที่ 2) คือ 0 และ 1 ตามลำดับ

ตัวอย่างการ run 1

Input n: 5
Fibonacci Seq: 0, 1, 1, 2, 3

ตัวอย่างการ run 2

Input n: 10
Fibonacci Seq: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Fibonacci Calculation

	ค่าของ พจน์ที่ $n-2$		ค่าของ พจน์ที่ $n-1$		ค่าของ พจน์ที่ n
n = 3	0	+	1	=	1
n = 4	1	← +	1	← =	2
n = 5	1	← +	2	← =	3
n = 6	2	← +	3	← =	5
n = 7	3	← +	5	← =	8