

# บทที่ 10 อาร์เรย์ (Array)



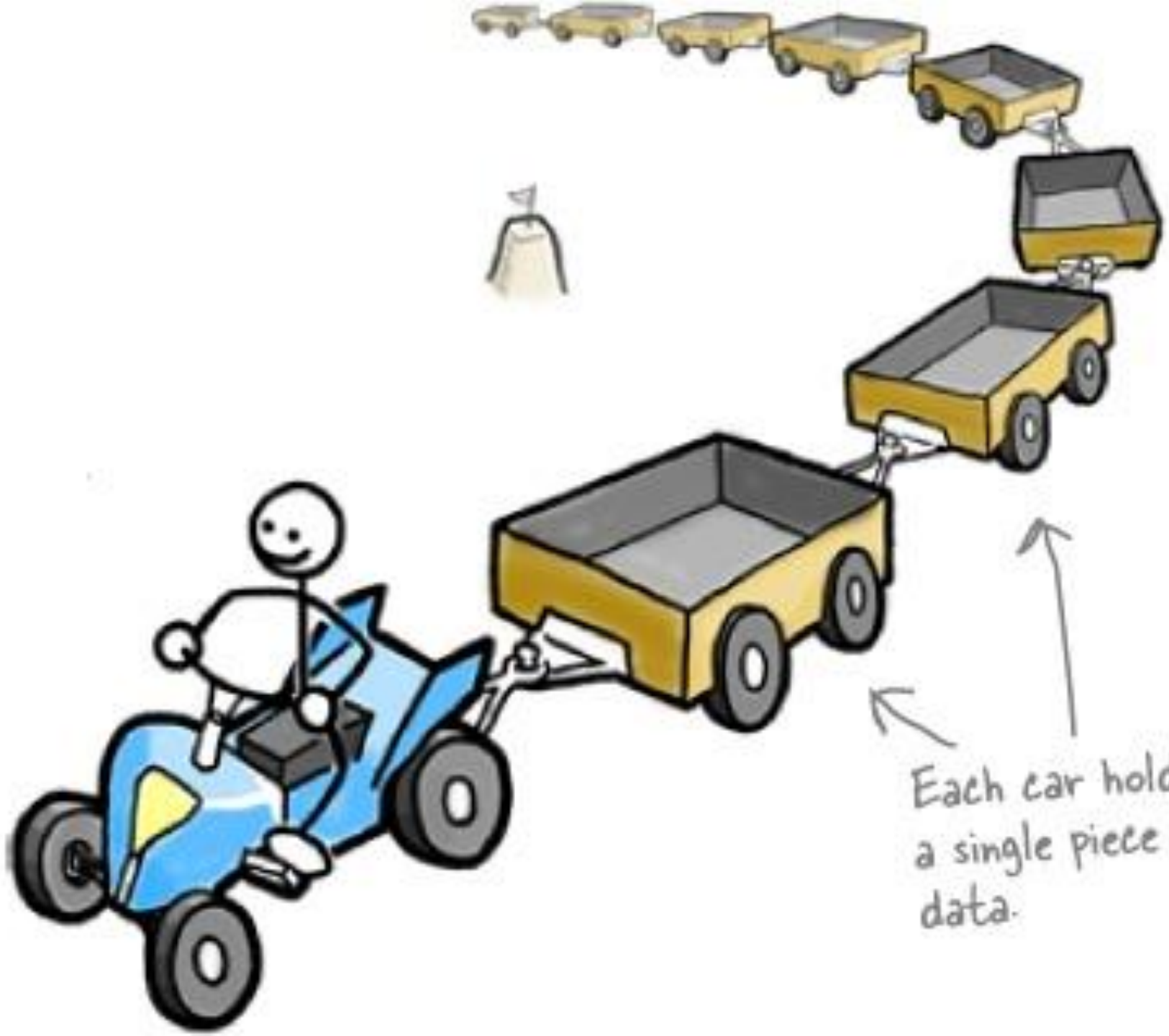
รายวิชา สธ 113 การออกแบบโปรแกรมทางธุรกิจเบื้องต้น

อ.อภิพงศ์ ปิงยศ

# Outline

- ▶ อาร์เรย์ของตัวเลข (Array of Numbers)
- ▶ อาร์เรย์ของตัวอักษร (Array of Characters : String))
- ▶ อาร์เรย์หลายมิติ (Multi-Dimensional Arrays)

Here comes the data train.

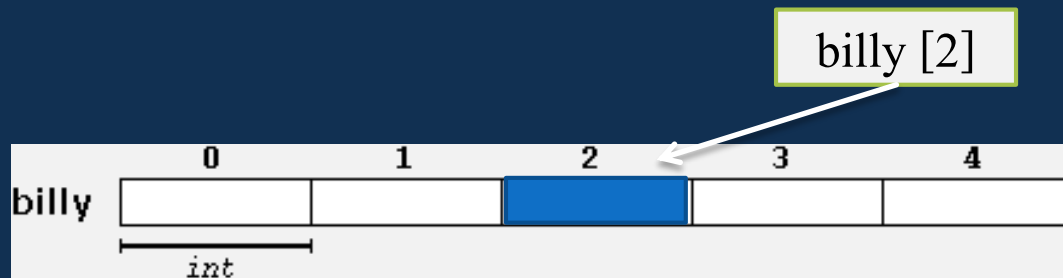


Each car holds a single piece of data.

Array of numbers

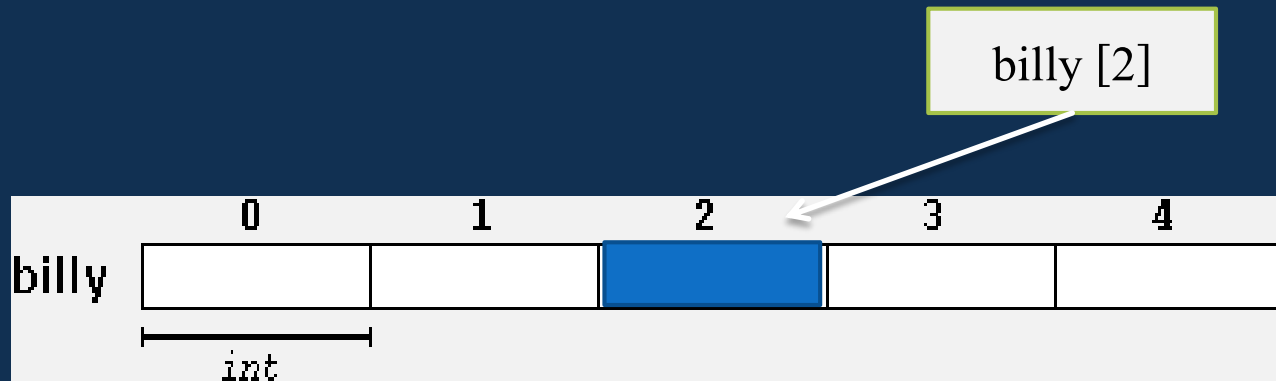
# ความหมายของอาร์เรย์ (Definition of Arrays)

- ▶ อาร์เรย์คือกลุ่มของตำแหน่งในหน่วยความจำที่เกี่ยวข้องกันโดยมีชื่อร่วมกันและมีชนิดเหมือนกัน
- ▶ ตัวอย่างเช่น ประกาศให้ตัวแปรชื่อว่า `billy` มีสมาชิกประเภท `int` 5 ตัวสามารถทำให้เราสามารถเก็บข้อมูลทั้ง 5 ตัวได้โดยใช้ชื่อเพียงชื่อเดียว



## ความหมายของอาร์เรย์ (Definition of Arrays) [ต่อ]

- ▶ แต่ละช่องว่างในรูปแสดงถึงสมาชิกในอาร์เรย์ โดยลำดับของสมาชิกมีตั้งแต่ 0 ถึง 4
- ▶ ลำดับแรกของสมาชิกอาร์เรย์นั้น คือ 0 เสมอ
- ▶ การที่จะอ้างถึงตำแหน่งของสมาชิก (index) ตัวใดในอาร์เรย์ ใช้ชื่ออาร์เรย์และหมายเลขระบุตำแหน่งหรือลำดับของสมาชิกในอาร์เรย์ อาทิเช่น



# การประกาศอาร์เรย์ (Declaration of Arrays)

- ▶ การจะใช้อาร์เรย์นั้นต้องมีการประกาศก่อนใช้เหมือนเช่นตัวแปรธรรมดา โดยปรกติอยู่ในรูปแบบ

```
type name[elements] ;
```

- ซึ่ง name คือ ชื่ออาร์เรย์
- type หมายถึงชนิด อาทิเช่น int, float, ฯลฯ และ elements ระบุจำนวนสมาชิกในอาร์เรย์
- โดยในบทเรียนนี้ elements จะเป็นค่าคงที่เท่านั้น

ตัวอย่าง

```
int billy[5] ;  
float c[12], x[27] ;
```

# การกำหนดค่าเริ่มต้นในอาร์เรย์ (Array Initialization)

▶ วิธีการที่สอง คือทำการเริ่มต้นค่าในโปรแกรมอาทิเช่น การกำหนดให้ค่าเริ่มต้นของสมาชิกทั้ง 5 ใน อาร์เรย์ billy มีค่าเป็น 0

ตัวอย่างที่ 1:

```
#include <stdio.h>
int i, billy[5];
void main()
{
    for ( i = 0; i < 5; i++ ) {
        billy[ i ] = 0; // set element at location i to 0
    }
}
```



# การกำหนดค่าเริ่มต้นในอาร์เรย์ (Array Initialization)

- ▶ ค่าของแต่ละสมาชิกในอาร์เรย์หลังจากการประกาศใช้ นั้นมักจะไม่ใช่ค่าที่เราต้องการ ดังนั้นเราควรมีการใส่ค่าเริ่มต้นไปในอาร์เรย์ โดยมีอยู่สองวิธีหลักๆ คือ
- ▶ **วิธีแรก** การกำหนดค่าเริ่มต้นพร้อมกับการประกาศ อาทิเช่น การกำหนดให้อาร์เรย์ **billy** มีค่าเป็นดัง

	0	1	2	3	4
billy	16	2	77	40	12071

```
int billy [5] = {16, 2, 77, 40, 12071 };
```

- ▶ นอกจากนั้นผลที่เกิดขึ้นสำหรับชุดคำสั่งในแต่ละชุดด้านล่างเป็นดังนี้

n มีสมาชิก 5 ตัว

```
int n[10] = { 1 };  
int n[] = {1, 2, 3, 4, 5};  
int n[5] = {32, 27, 64, 18, 95, 14};
```

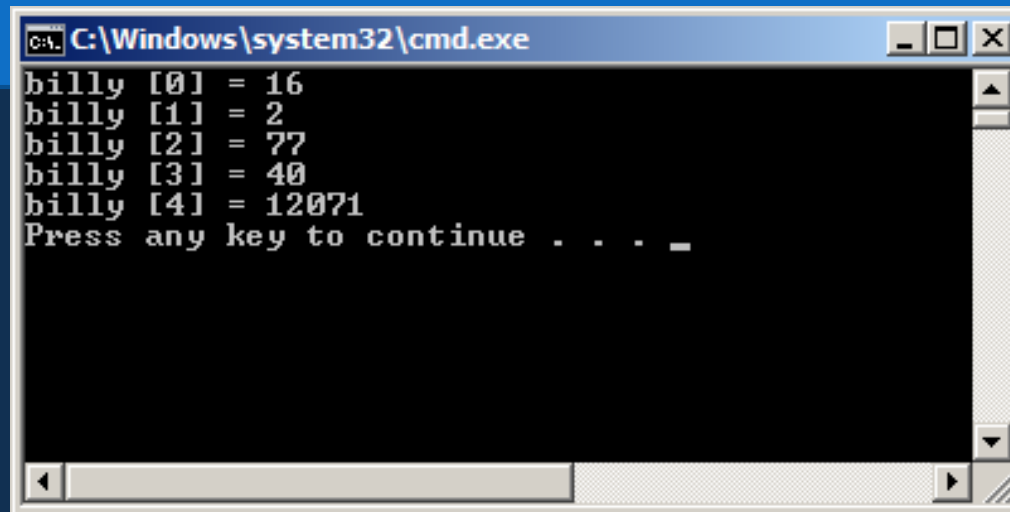
{1, 0, 0, ..., 0}

Error!

## ตัวอย่างที่ 2

- ▶ แสดงค่าของสมาชิกในอาร์เรย์ **billy**

```
#include <stdio.h>
int i, billy[5] = {16, 2, 77, 40, 12071};
void main()
{
    for ( i = 0; i < 5; i++ ) {
        printf("billy [%d] = %d \n",i, billy[i]);
    }
}
```

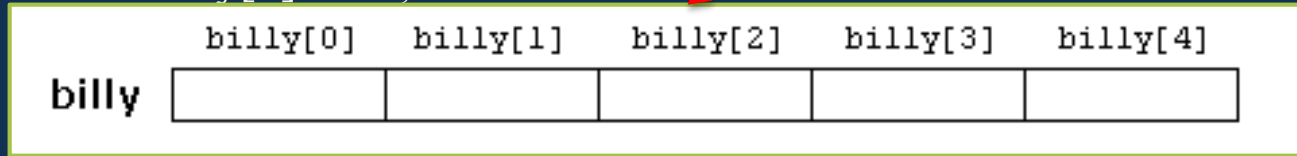


```
C:\Windows\system32\cmd.exe
billy [0] = 16
billy [1] = 2
billy [2] = 77
billy [3] = 40
billy [4] = 12071
Press any key to continue . . . _
```

# การใช้ค่าต่างๆในอาเรย์

- ▶ รูปแบบการใช้ค่าต่างๆในอาเรย์คือ `name[index]`

ตัวอย่างเช่น `billy[2] = 75;`



- ▶ กำหนดให้ สมาชิกลำดับที่ 3 (แต่ค่า `index = 2`) ใน **billy** มีค่าเป็น 75

```
a = billy[2];
```

และ

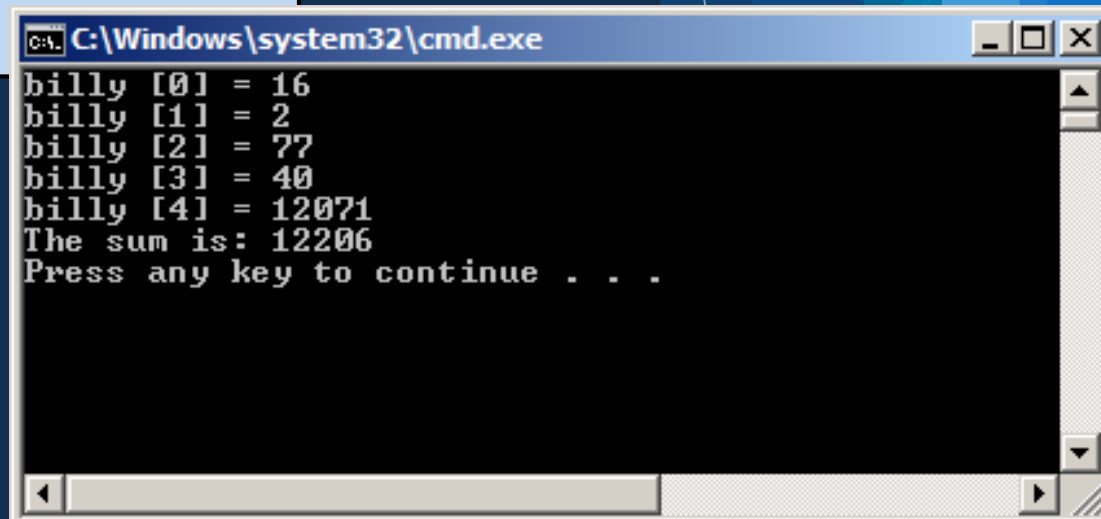
เป็นการผ่านค่าดังกล่าวไปยังตัวแปร **a**

## ข้อควรระวังเมื่อใช้อาเรย์

เนื่องจากอาเรย์ใน ภาษา C นั้นไม่มีการตรวจสอบขอบเขตของอาเรย์ ดังนั้นเราสามารถทำข้อผิดพลาดโดยการอ้างอิงขอบเขตที่กำหนดไว้ได้ อาทิ เช่น `billy[5]` เป็นต้น

## ตัวอย่าง 3: การใช้อาเรย์เพื่อหาผลรวม

```
#include <stdio.h>
int i, billy[5] = {16, 2, 77, 40, 12071};
int billy_sum = 0;
void main()
{
    for ( i = 0; i < 5; i++ ) {
        printf("billy [%d] = %d \n",i,billy[i]);
        billy_sum = billy_sum + billy[i];
    }
    printf("The sum is: %d \n",billy_sum);
}
```



```
C:\Windows\system32\cmd.exe
billy [0] = 16
billy [1] = 2
billy [2] = 77
billy [3] = 40
billy [4] = 12071
The sum is: 12206
Press any key to continue . . .
```

# การรับค่าของอาร์เรย์ (Input of Array Values)

ในกรณีที่กำหนดค่าตัวแปรอาเรย์จากผู้ใช้ผ่านคีย์บอร์ดสามารถทำได้โดยใช้ร่วมกับคำสั่ง **scanf()** เช่นกัน

```
scanf ("%d %f", &grades [0] , &price [2]) ;  
scanf ("%d" , &code [0]) ;  
scanf ("%d %d %d",  
&grades [0] , &grades [1] , &grades [2]) ;
```

# การรับค่าของอาร์เรย์ (Input of Array Values)

หรือการวนรอบเพื่อรับค่าจากผู้ใช้เป็นไปดังตัวอย่างนี้

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

จะเป็นการวนรอบรับค่าจำนวนเต็มจากผู้ใช้งานจำนวน 5 ค่าและเก็บไว้ในอาร์เรย์ grades

## ตัวอย่าง4: การใช้อาเรย์เพื่อรับค่าอินพุตหลายค่า

```
#include <stdio.h>
int i;
float billy[5];
void main()
{
    for(i = 0; i < 5; i++) {
        printf("Enter a test score: " );
        scanf("%f", &billy[i]);
    }
}
```

# Arrays of Characters (String)



# การใช้งานอาร์เรย์ของตัวอักษร (Strings)

- ▶ การใช้อาร์เรย์นั้นไม่ได้ถูกจำกัดอยู่เพียงข้อมูลที่เป็นตัวเลขเท่านั้น ยังสามารถนำมาใช้ได้กับตัวอักษร หรือ string อาทิเช่น

```
char string1[] = "first";
```

หรือ

```
char string1[] = {'f', 'i', 'r', 's', 't', '\0'};
```

โดยที่ string1 ประกอบไปด้วยตัวอักษร 5 ตัวและ ตัวอักษรพิเศษที่ใช้ในการสิ้นสุดสตริงเรียกว่า ตัวอักษรนัล (Null Character) ดังนั้นอาร์เรย์ string1 จึงประกอบด้วย **สมาชิกจำนวนหกตัว** โดยใช้ '\0' สำหรับตัวอักษรนัล สตริงทุกตัวในภาษาซีจะต้องปิดท้ายด้วยตัวอักษรนัลเสมอ

## การใช้ String สำหรับอินพุต

- เราสามารถที่จะรับค่าอินพุตสตริงจากคีย์บอร์ดแล้วเก็บในอาเรย์ของตัวอักษรได้โดยใช้ scanf และการใช้การกำหนดรูปแบบ %s ตัวอย่างเช่น ประกาศอาเรย์ตัวอักษร `char string2[12];` และรับข้อมูล `scanf( "%s", string2 );` จะใช้เพื่อเก็บตัวอักษร ไม่เกิน 11 ตัวจากคีย์บอร์ด และเก็บไว้ในตัวแปร `string2`

การใช้ string สำหรับอินพุตนั้น ใช้ control character คือ %s และความยาวของอินพุตนั้นต้องไม่มากกว่า ขนาดของอาเรย์

## การใช้ String สำหรับอินพุต

```
char string2[12]  
scanf( "%s", string2 );
```

นอกจากนี้จะสังเกตได้ว่า string2 นั้นถูกใช้โดยไม่ต้องมีเครื่องหมาย & นำหน้า ดังเช่นในอินพุตประเภทอื่นๆ เพราะ string2 นั้นระบุตำแหน่งของข้อมูลไว้เรียบร้อยแล้ว

เนื่องจาก string2 เป็นตัวแปรประเภท string ดังนั้น เมื่อใช้ scanf ไม่มีความจำเป็นต้องใช้ เครื่องหมาย & นำหน้าชื่อตัวแปร

# ตัวอย่าง การรับค่าสตริง November

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

```
void main()
```

```
{
```

```
    char string2[12];
```

```
    printf("Please enter a string:");
```

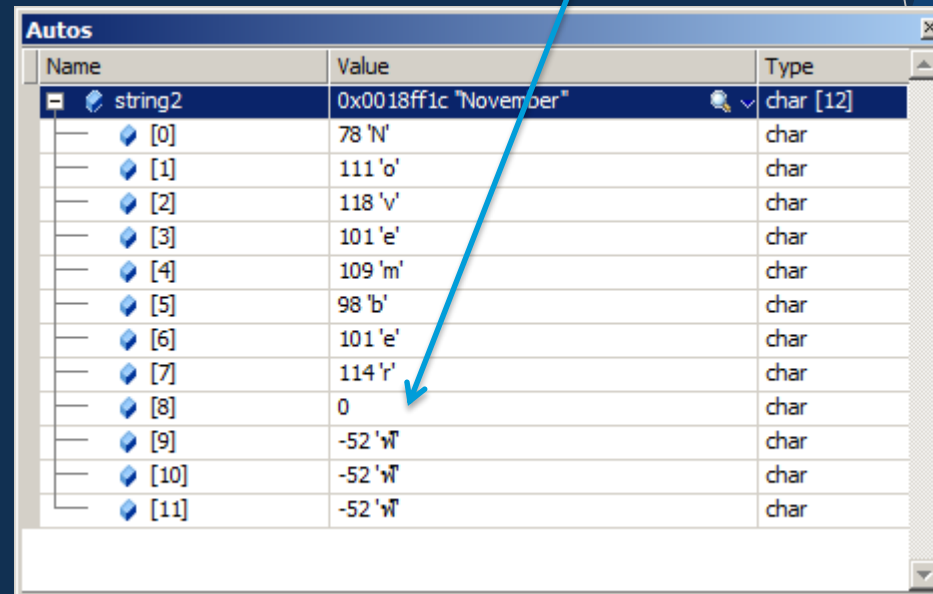
```
    scanf("%s", string2);
```

```
    printf("\n The string is called: %s\n", string2);
```

```
}
```

สิ้นสุดด้วย ค่า 0 ณ  
ตำแหน่งของ string2[8]

ตัวอย่างแสดงให้เห็นถึง ค่าของ ตัวแปร string2 ในแต่ละตำแหน่ง พร้อมทั้งค่า ASCII อาทิเช่น 78 สำหรับอักขระ 'N' หรือ 109 สำหรับอักขระ 'm' และ มีการลงท้ายสตริง ด้วย ตัวอักษร null ซึ่งมีค่าเป็น 0 อยู่ด้วย



Name	Value	Type
string2	0x0018ff1c "November"	char [12]
[0]	78 'N'	char
[1]	111 'o'	char
[2]	118 'v'	char
[3]	101 'e'	char
[4]	109 'm'	char
[5]	98 'b'	char
[6]	101 'e'	char
[7]	114 'r'	char
[8]	0	char
[9]	-52 'ฟ'	char
[10]	-52 'ฟ'	char
[11]	-52 'ฟ'	char

## Example

- ▶ จงเขียนโปรแกรมที่อ่านข้อมูลจากคีย์บอร์ด และพิมพ์ข้อมูลนั้นกลับออกไปแบบย้อนกลับ อาทิเช่น เมื่ออินพุตคือ

November

เอาท์พุตควรเป็น

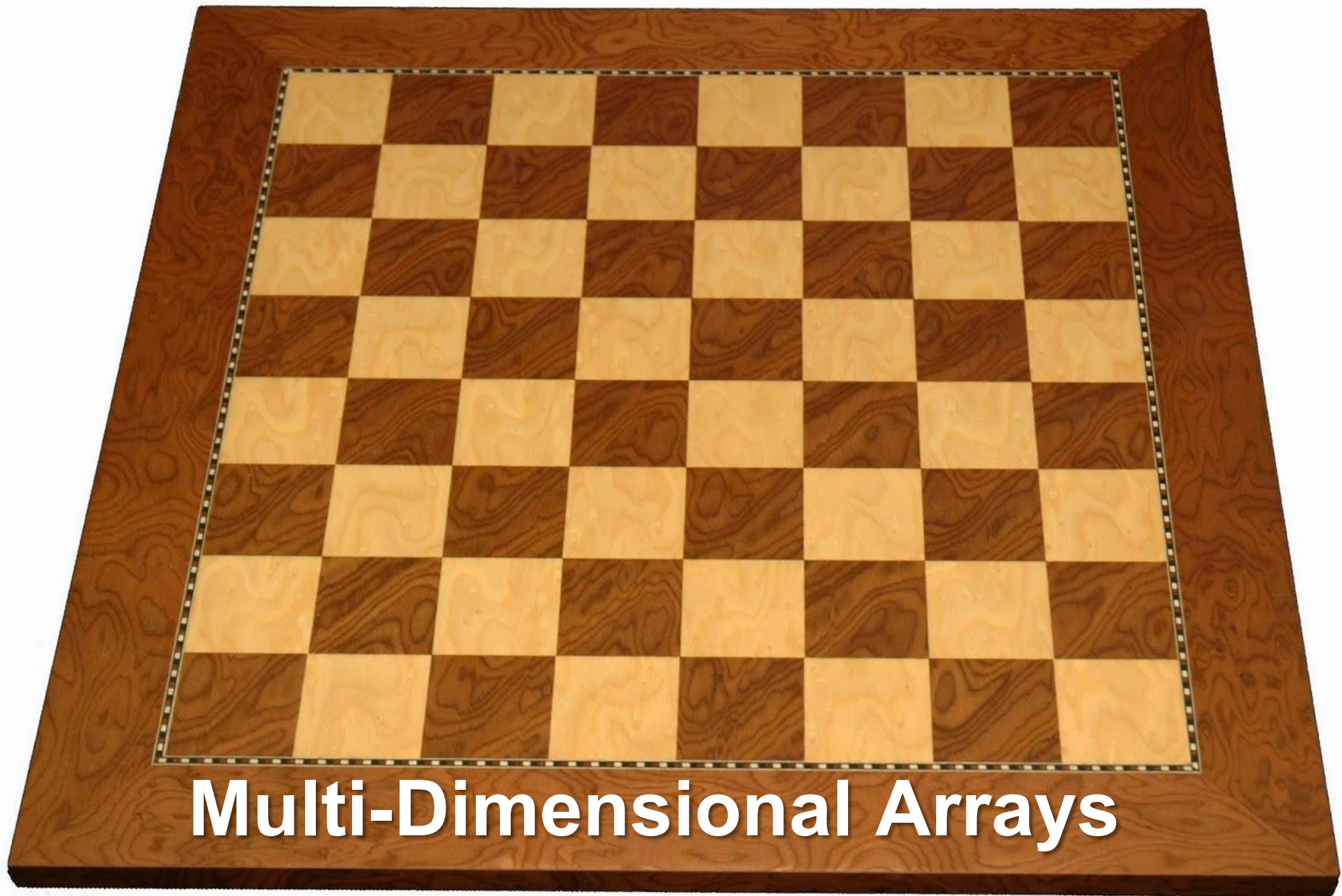
rebmevoN

สมมุติฐานคือ ข้อมูลมีความยาวไม่เกิน 12 ตัวอักษร

หมายเหตุ เราสามารถหาความยาวของข้อความประเภท String ได้โดยใช้ Function ที่ชื่อว่า `strlen()` ซึ่งอยู่ใน Library ชื่อว่า `string.h`

# Example

```
#include <stdio.h>
#include <string.h>
void main()
{
    char string2[12];
    int i;
    printf("Please enter a string:");
    scanf("%s",string2);
    /* Now print it in reverse */
    for (i = strlen(string2) - 1 ; i>=0; i--) {
        printf("%c", string2[i]);
    }
    printf("\n");
}
```



# Multi-Dimensional Arrays

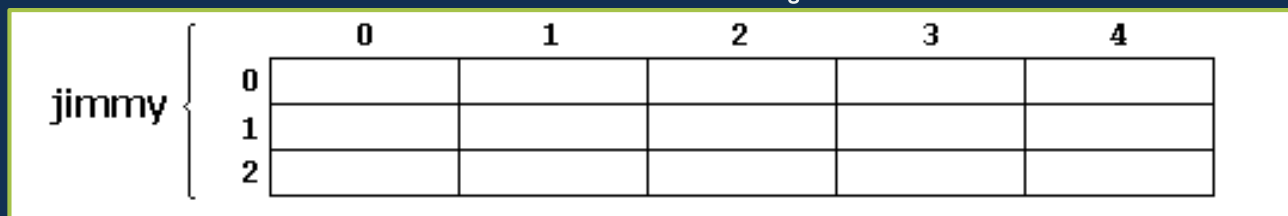
# อาร์เรย์หลายมิติ

- ▶ อาร์เรย์ในภาษาซีสามารถมีได้หลายมิติ โดยในที่นี้เราจะศึกษาอาร์เรย์แบบ 2 มิติ ซึ่ง

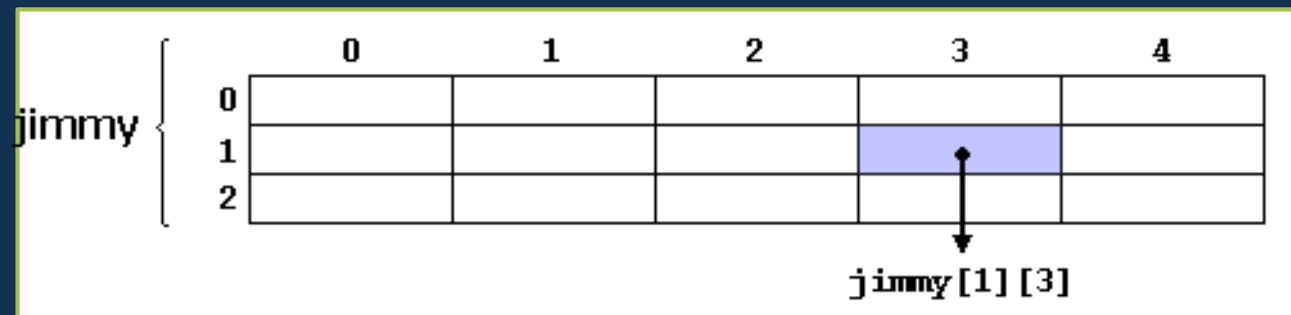
การประกาศ

```
int jimmy[3][5]
```

หมายถึง อาร์เรย์ ประเภท int ที่มีขนาด 3 แถว x 5 คอลัมน์ ดังรูป



ดังนั้นข้อมูล jimmy[1][3] หมายถึงข้อมูลแถวที่สองจากด้านบนและคอลัมน์ที่สี่จากซ้ายมือ ดังรูป





# การกำหนดค่าในอาเรย์สองมิติ

การกำหนดค่าเริ่มต้นในตัวแปรอาร์เรย์ 2 มิติสามารถกระทำได้โดยจะเพิ่มส่วนการแบ่งแยกในแต่ละแถวด้วยเครื่องหมาย “{ }” และ “,” เช่น

```
int jimmy[3][5] = {{1,2,3,4,5},  
                  {2,4,6,8,10},  
                  {3,6,9,12,15}};
```

การให้ค่าเริ่มต้นแก่ตัวแปรอาร์เรย์ 2 มิติยังสามารถที่จะละเครื่องหมายปีกกาในแต่ละแถวได้โดยให้ใช้เครื่องหมายจุลภาคแทนเช่น

```
int jimmy[3][5] = {1,2,3,4,5,  
                  2,4,6,8,10,  
                  3,6,9,12,15};
```

หรือสามารถเขียนได้เป็น

```
int jimmy[3][5] = {1,2,3,4,5,2,4,6,8,10,3,6,9,12,15};
```

# ตัวอย่างการใช้ อาร์เรย์สองมิติ

```
#define WIDTH 5
#define HEIGHT 3

int jimmy[HEIGHT][WIDTH];
int n,m;
int main()
{
    for (n=0;n<HEIGHT;n++)
        for (m=0;m<WIDTH;m++) {
            jimmy[n][m]=(n+1)*(m+1);
        }
    return 0;
}
```

ตารางแสดงค่าที่เก็บไว้ใน  
แต่ละตำแหน่งในอาร์เรย์  
jimmy

	0	1	2	3	4
0	1	2	3	4	5
1	2	4	6	8	10
2	3	6	9	12	15

# โปรแกรมหาผลรวมของสมาชิกในอาเรย์

```
total = 0;  
for (n= 0; n< HEIGHT; n++ )  
  for ( m = 1; m <= WIDTH; m++ )  
  {  
    total = total + jimmy[n][m];  
  }
```

	0	1	2	3	4
0	1	2	3	4	5
1	2	4	6	8	10
2	3	6	9	12	15