

ฟังก์ชันส่งผ่านอาร์เรย์

ENGCE115 Computer Programming for Computer Engineer

เนื้อหา

1. ทบทวนเรื่อง Pointer
2. Pointer Array
3. Pointer Function
4. Dynamic Array

ทบทวนเรื่อง Pointer

ทบทวนเรื่อง Pointer

- Pointer คือตัวชี้ ซึ่งถือเป็นลักษณะเด่นของ Operator ในภาษา C
- มีประโยชน์ในการจัดการข้อมูลแบบ Dynamic data structure
- ตัวแปรชนิด Pointer จะเก็บค่า “ที่อยู่” ของหน่วยความจำ (แตกต่างกับตัวแปรชนิดอื่นที่เก็บแค่ค่าจริง)
- รูปแบบการประกาศตัวแปรคือ

ชนิดข้อมูล *ชื่อตัวแปร

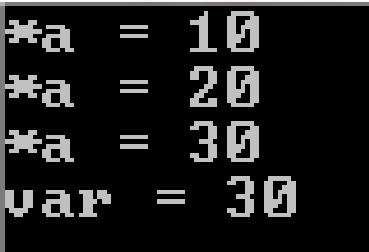
เช่น `int *variable;`

ชนิดข้อมูล* ชื่อตัวแปร

เช่น `int* variable;`

ทบทวนเรื่อง Pointer

```
1  #include <stdio.h>
2
3  int main(){
4      int *a;
5      int var = 10;
6      a = &var;
7
8      printf("*a = %d\n", *a);
9
10     var = 20;
11     printf("*a = %d\n", *a);
12
13     *a = 30;
14     printf("*a = %d\n", *a);
15     printf("var = %d\n", var);
16
17     return 0;
18 }
```



```
*a = 10
*a = 20
*a = 30
var = 30
```

Pointer Array

Array เก็บ Pointer 3 ตัว

```
1  #include <stdio.h>
2
3  int main(){
4      int *a[3]; // *(a[3])
5      int x, y, z;
6      x = 1; y = 2 ; z = 3;
7      a[0] = &x;
8      a[1] = &y;
9      a[2] = &z;
10     printf("*a[0] = %d\n", *a[0] );
11     printf("*a[1] = %d\n", *a[1] );
12     printf("*a[2] = %d\n", *a[2] );
13     printf("\n");
14     printf("a[0] = %d (&x = %d)\n", a[0], (&x) );
15     printf("a[1] = %d (&y = %d)\n", a[1], (&y) );
16     printf("a[2] = %d (&z = %d)\n", a[2], (&z) );
17     return 0;
18 }
```

```
*a[0] = 1
*a[1] = 2
*a[2] = 3

a[0] = 2293084 (&x = 2293084)
a[1] = 2293080 (&y = 2293080)
a[2] = 2293076 (&z = 2293076)
```

Pointer 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 3 Column

การประกาศใช้

`int (*a)[3];` หรือ `a = new int[n][3];`

ค่าที่ได้คือ

	0	1	2
a[0]	Data	Data	Data
a[1]	Data	Data	Data
:	:	:	:
a[0]	Data	Data	Data

Pointer 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 3 Column

```
1  #include <stdio.h>
2
3  int main(){
4      int n = 5;
5      int g[3][5] = {
6          { 10, 20, 30, 40, 50 } ,
7          { 11, 21, 31, 41, 51 } ,
8          { 12, 22, 32, 42, 52 }
9      };
10
11     for( int a = 0 ; a < 3 ; a++ ){
12         for( int b = 0 ; b < 5 ; b++ )
13             printf("%d ", g[a][b]);
14         printf("\n");
15     }
16
17     printf("\n-----\n");
18
19     int (*j)[5] = g;
20
21     j[1][4] = 99;
22
23     for( int a = 0 ; a < 3 ; a++ ){
24         for( int b = 0 ; b < 5 ; b++ )
25             printf("%d ", g[a][b] );
26         printf("\n");
27     }
28     return 0;
29 }
```

10	20	30	40	50
11	21	31	41	51
12	22	32	42	52

10	20	30	40	50
11	21	31	41	99
12	22	32	42	52



Pointer Function

Reference

ตัวแปร

B



*A



**p

Address

Value

1000

10

1001

1002

1000

1003

1004

1002

1005

1006

Reference

```
1  #include <stdio.h>
2
3  [ ] int main(){
4      int a = 10;
5      int *b = &a;
6      int **p = &b;
7
8      printf("  %d", **p);
9
10     return 0;
11 }
```

A digital display with a black background and white pixelated text showing the number "10".

ดังนั้น $*a === **p$

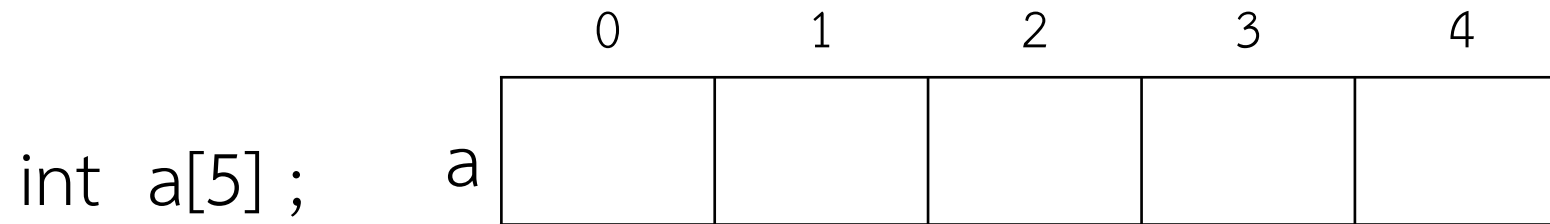
Pointer Function

```
3 ☐ int add( int x, int y ){  
4   |   return x + y;  
5   |   }  
6  
7 ☐ int operation( int x, int y, int (*function) (int, int) ){  
8   |   return (*function)(x, y);  
9   |   }  
10  
11 ☐ int main(){  
12   |   printf("==> %d\n", operation(5, 5, add) );  
13   |   return 0;  
14   |   }  
15
```

==> 10

Dynamic Array

ที่มา



int *b ;

หมายเหตุ : b ก็เป็น Array ได้เหมือนกัน เรียกว่า Dynamic Array

ที่มา

- เพราะไม่รู้ว่าต้องการใช้เท่าไร
- จะประกาศล่วงหน้าเยอะๆ ก็ไม่ดี
- ดังนั้น Pointer จึงเป็นทางออก
- เอาไปทำเป็น Array ที่มีสมาชิกกี่ตัวก็ได้ โดยการเอาไปจองหน่วยความจำ
- วิธีการมีดังนี้

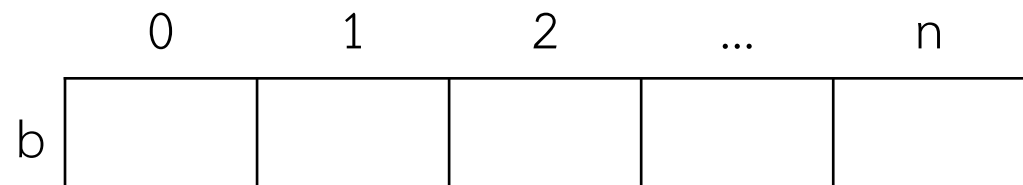
Dynamic Array

```
int *b;
```

```
b = new int[n];
```



เอา b ไปจอง Address



Dynamic Array

- ดังนั้น หากต้องการจอง 2 มิติก็ต้องทำลักษณะนี้ ?

```
int **c ;
```

```
c = new int [n][m] ;
```



Dynamic Array

- การจอง 2 มิติ ที่ถูกต้องควรทำในลักษณะนี้

```
int (*c)[4];
```

- ข้อสังเกตคือ ตรงไหนที่มี Pointer หมายความว่ายืดหยุ่นได้

```
int **c;
```

```
c = new int [n][4];
```



ลักษณะการทำงาน

สมมติเรามี Array 2 มิติ `int a[3][4];`

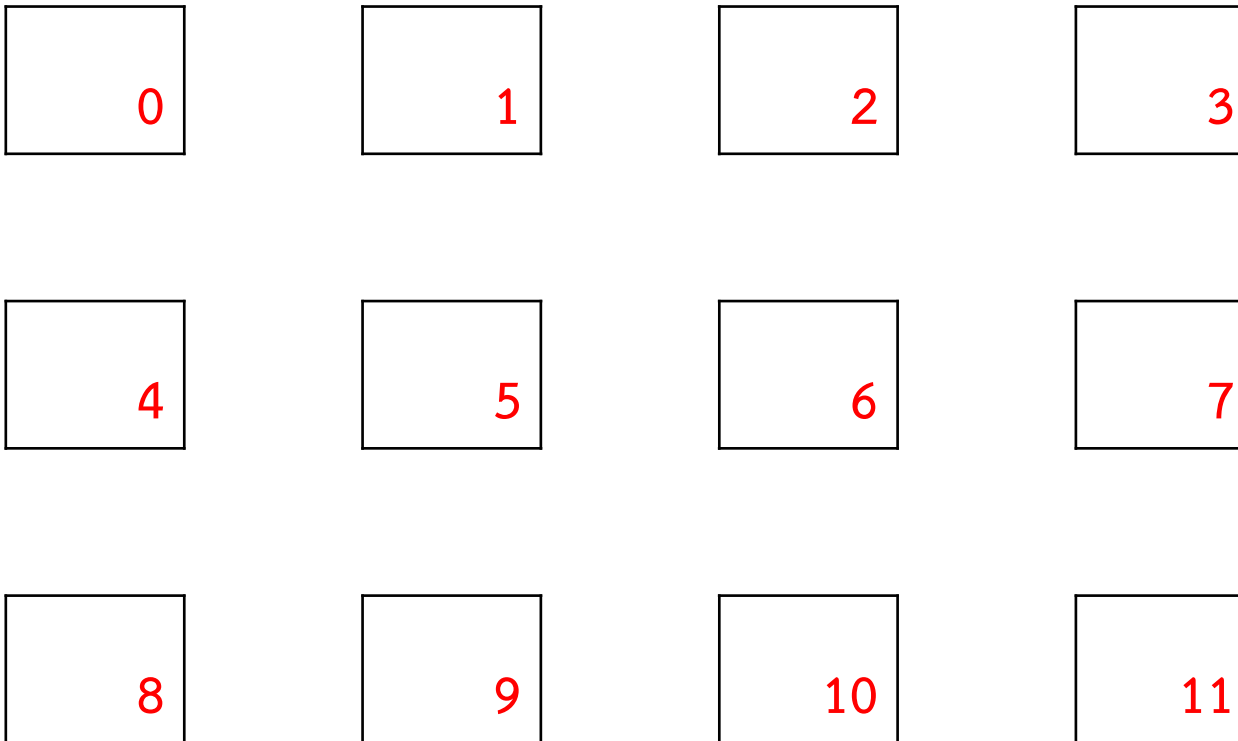
← j ตัวบอก Column

	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11

↓ i ตัวบอก Row

ลักษณะการทำงาน

สร้าง 1 มิติ เป็น 2 มิติ โดยการตัด Array แล้วเอามาต่อกัน



ลักษณะการทำงาน

นั่นคือเอา Array ขนาด 12 มาแทน Array 3x4

0	1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	---	----	----

การสร้างคือ

```
int *a;
```

```
a = new int [ 3 x 4 ];
```

ลักษณะการทำงาน

แบบ 2D

$a[1][2]$

$a[i][j]$

	0	1	2	3
0				
1				
2				

แบบ 1D

$a[6] = a[1 \times 4 + 2]$

$a[i \times \text{col} + j]$

0	1	2	3	4	5	6	7	8	9	10	11

ลองเขียนโปรแกรม

ตัวอย่างที่ 1 Dynamic Array (การจองและการใช้งาน)

```
1  #include <stdio.h>
2  int main(){
3      int row, col, i, j;
4      printf("How many rows, cols?");
5      scanf("%d %d", &row, &col);
6
7      int a[row][col];
8
9      for(int i = 0 ; i < row ; i++)
10         for(int j = 0 ; j < col ; j++){
11             printf("a[%d][%d]: ", i, j);
12             scanf( "%d", &a[i][j] );
13         }
14
15     return 0;
16 }
```


ลองเขียนโปรแกรม

ตัวอย่างที่ 1 Dynamic Array (การจองและการใช้งาน)

```
1  #include <stdio.h>
2  int main(){
3      int row, col, i, j;
4      printf("How many rows, cols?");
5      scanf("%d %d", &row, &col);
6
7      int a[row][col]; Error ที่บรรทัดนี้เพราะประกาศแบบนี้ไม่ได้
8
9      for(int i = 0 ; i < row ; i++)
10         for(int j = 0 ; j < col ; j++){
11             printf("a[%d][%d]: ", i, j);
12             scanf( "%d", &a[i][j] );
13         }
14
15     return 0;
16 }
```

รับค่าแบบนี้ไม่ได้ ต้องแก้ด้วย

ลองเขียนโปรแกรม

ตัวอย่างที่ 1 Dynamic Array (การจองและการใช้งาน)

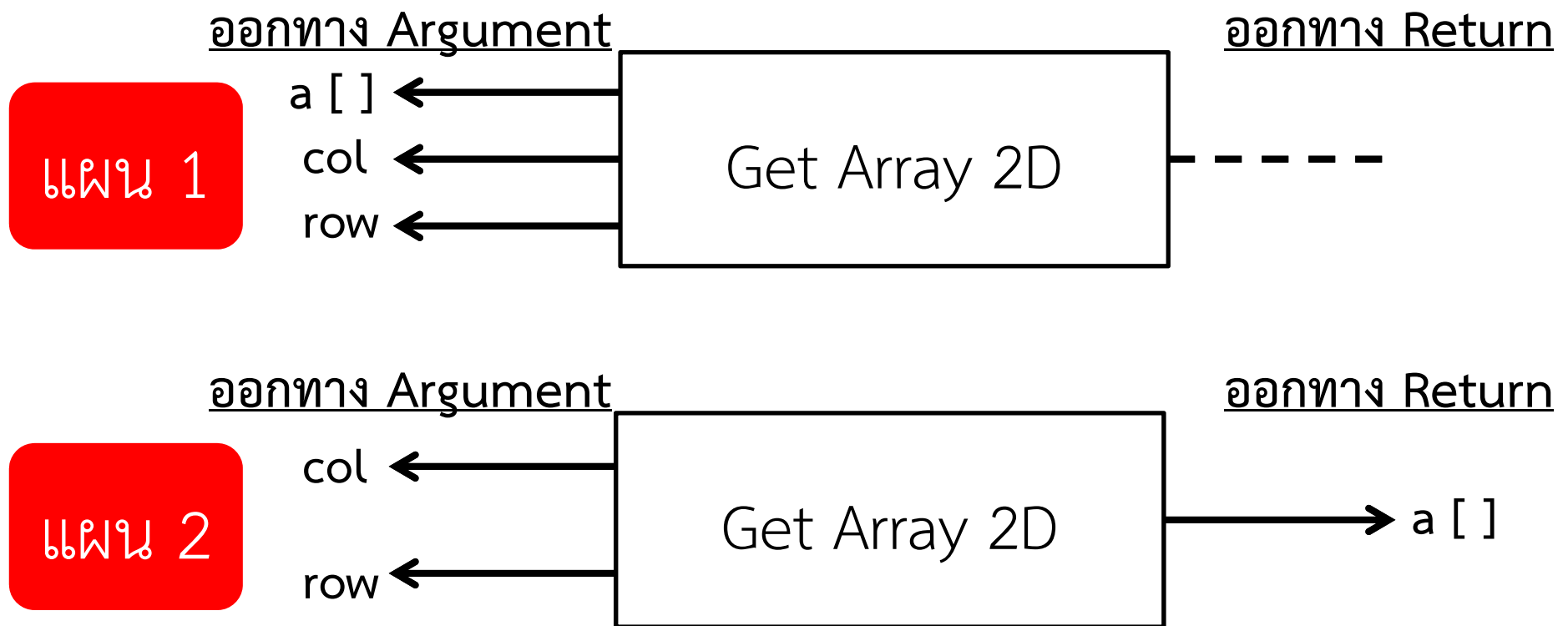
```
1  #include <stdio.h>
2  int main(){
3      int row, col, i, j;
4      printf("How many rows, cols?");
5      scanf("%d %d", &row, &col);
6
7      int *a;
8      a = new int[ row * col ]; ประกาศ a เป็น Pointer ซะเลย
9
10     for(int i = 0 ; i < row ; i++)
11     {
12         for(int j = 0 ; j < col ; j++){
13             printf("a[%d][%d]: ", i, j);
14             scanf( "%d", &a[ i * col + j ] );
15         }
16     }
17     return 0;
}
```

ลองเขียนโปรแกรม

- ในตัวอย่างที่ 1 นั้น Code ยังคงดูวุ่นวายอยู่
- ดังนั้นเราต้องแยกงานเฉพาะอย่าง ให้เป็น Function จะดีกว่า เช่น ส่วนของการเก็บข้อมูล

ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function



ลองเขียนโปรแกรม

ข้อสังเกต:

- Array ภายนอก เป็นเหมือน Pointer 1 ระดับอยู่แล้ว
- ดังนั้นการจะเอาข้อมูลออกจะต้องมีมากกว่าเดิมอยู่ 1 ระดับ



ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function

แผ่น 1

```
3 void GetArr2D( int **a, int *row, int *col ){
4     printf("How many row, col ? ");
5     scanf("%d %d", row, col);
6     *a = new int[ *row * *col ];
7     for(int i = 0 ; i < *row ; i++)
8     {
9         for(int j = 0 ; j < *col ; j++){
10             printf("a[%d][%d] = ", i, j);
11             scanf("%d", &(*a)[ i * *col + j ]);
12         }
13     }
```

เพราะต้องการเข้าใช้ตัวแปร arr ที่อยู่ใน main จึงต้องใช้ *a

ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function

แผน 1

Main

```
27 void GetArr2D( int **a, int *row, int *col );
28
29 int main(){
30     int r, c, *arr;
31     GetArr2D( &arr, &r, &c );
32     return 0;
33 }
```

ต้อง Prototype ไว้ก่อน

หมายเหตุ: ประโยชน์ของ Function คือสามารถเรียกใช้คำสั่งเดิมๆ โดยเปลี่ยนแค่ตัวแปรเท่านั้น

ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function

แผน 1

Function

Main

a	===	&arr
*a	===	arr
row	===	&r
*row	===	r



ใน Function เก็บ
Address ของ arr

ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function

แผน 2

```
14 void *GetArr2D( int *row, int *col ){
15     int *a;
16     printf("How many row, col ? ");
17     scanf("%d %d", row, col);
18     a = new int[ *row * *col ];
19     for(int i = 0 ; i < *row ; i++)
20         for(int j = 0 ; j < *col ; j++){
21             printf("a[%d][%d] = ", i, j);
22             scanf("%d", &a[ i * *col + j ]);
23         }
24     return a;
25 }
```

ประกาศขึ้นมาชั่วคราว

ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function

แผน 2

```
14 void *GetArr2D( int *row, int *col ){
15     int *a;
16     printf("How many row, col ? ");
17     scanf("%d %d", row, col);
18     a = new int[ *row * *col ];
19     for(int i = 0 ; i < *row ; i++)
20         for(int j = 0 ; j < *col ; j++){
21             printf("a[%d][%d] = ", i, j);
22             scanf("%d", &a[ i * *col + j ]);
23         }
24     return a;
25 }
```

ดังนั้นจึงใช้มันได้เลย
(ไม่เหมือนแผน 1)

Return a ได้เลย

ลองเขียนโปรแกรม

ตัวอย่างที่ 2 ลด Code โดยการแยกเป็น Function

แผน 2

Main

```
15  int *GetArr2D( int *row, int *col );
16
17  int main(){
18      int r, c, *arr;
19      arr = GetArr2D( &r, &c );
20      delete [] arr;
21      return 0;
22  }
```

ต้องลบหน่วยความจำด้วย