

อาร์เรย์ (ARRAY)

Nattapon Buaurai

22/02/2022

มาเล่นเกมกัน...

Wordle

???

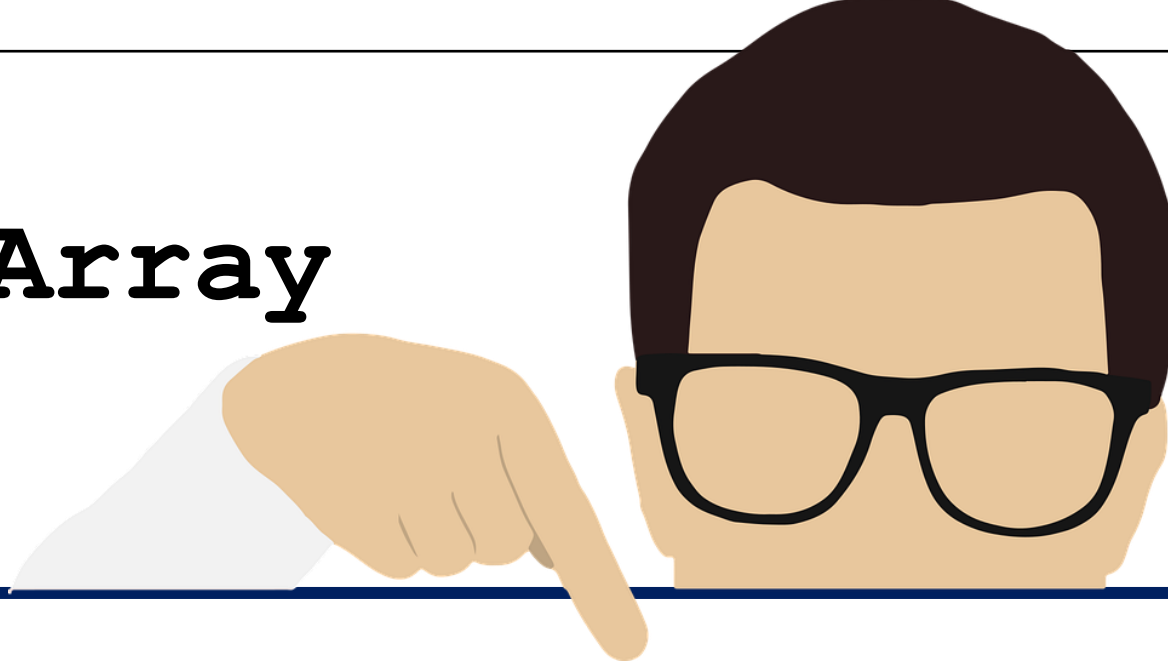
เขียนโปรแกรมเก็บค่าคะแนนนักเรียน
ทุกคนในห้องเรียน (45 คน)

Solve

main.c ×

```
1  #include <stdio.h>
2  int main() {
3      int score1, score2, score3, score4,
4      score5, score6, score7, score8, ...,
5      score45;
6
7      scanf("%d", &score1);
8      scanf("%d", &score2);
9      ...
10     scanf("%d", &score45);
11     return 0;
12 }
```

Array

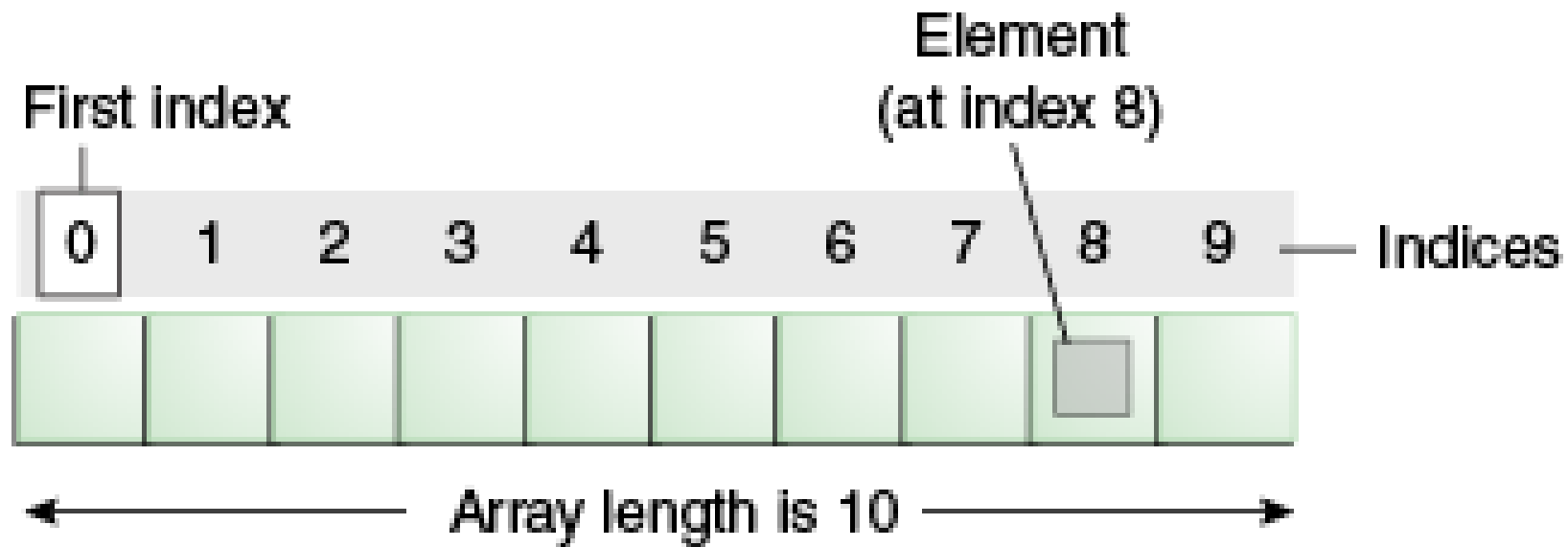


- ตัวแปรอาร์เรย์ เป็นตัวแปรประเภทหนึ่งที่ใช้ชื่อตัวแปรชื่อเดียวแต่สามารถเก็บข้อมูลชนิดเดียวกันเป็นกลุ่มได้

สิ่งที่ต้องรู้เกี่ยวกับอาร์เรย์

- อาร์เรย์จะทำหน้าที่จองเนื้อหามหน่วยความจำตามขนาดที่ระบุ
- ข้อมูลแต่ละตัวเรียกว่า Element หรือ Cell
- การอ้างถึงข้อมูลแต่ละ Cell จะใช้ Index เป็นตัวชี้

Array



การประกาศตัวแปร **Array**

```
1  #include <stdio.h>
2  int main() {
3      int score[45];
4      float gpa[45];
5
6      return 0;
7  }
```

จำนวนสมาชิกของตัวแปรอาร์เรย์



ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์[n];



ประเภทข้อมูลในตัวแปรอาร์เรย์

ถ้าหากประกาศตัวแปรเป็นสตริง ตัวแปรนั้นก็คืออาร์เรย์ของ char นั่นเองครับ

การกำหนดค่าให้ตัวแปร **Array**

```
1  #include <stdio.h>
2  int main() {
3      int score[45];
4      float gpa[45];
5      score[0] = 50;
6      score[20] = 75;
7      gpa[1] = 3.98;
8      return 0;
9  }
```

จะเกิดอะไรขึ้น ถ้ากำหนดค่า
เกินขอบเขตของอาร์เรย์ที่ประกาศไว้

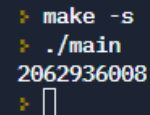
การกำหนดค่าให้ตัวแปร **Array**

```
1  #include <stdio.h>
2  int main() {
3      int score[45];
4      float gpa[45];
5      score[50] = 98;
6      printf("%d\n", score[50]);
7      return 0;
8  }
```



```
C:\Users\Joke-ICT\Desktop\test1.exe
98
Process returned -1073741819 (0xC0000005)  execution time : 0.467 s
Press any key to continue.
```

```
1  #include <stdio.h>
2  int main() {
3      int score[45];
4      float gpa[45];
5      score[50] = 98;
6      printf("%d\n", score[50]);
7      return 0;
8  }
```



```
> make -s
> ./main
2062936008
> 
```

การกำหนดค่าให้ตัวแปร **Array** ด้วย **Loop**

```
1  #include <stdio.h>
2  int main() {
3      int i, score[45];
4      for(i=0; i<=44; i++)
5      {
6          printf("Input score %d : ", i);
7          scanf("%d", &score[i]);
8      }
9      return 0;
10 }
11
```

C:\Users\Joke-ICT\Desktop\test1.exe

```
Input score 0 : 1
Input score 1 : 2
Input score 2 : 3
Input score 3 : 4
Input score 4 : 5
Input score 5 : _
```

การอ้างอิงสมาชิกใน **Array**

```
1  #include <stdio.h>
2  int main() {
3      int i, score[5];
4      for(i=0; i<=4; i++)
5      {
6          printf("Input score %d : ", i);
7          scanf("%d", &score[i]);
8      }
9      printf("Score[%d] = %d\n", 0, score[0]);
10     printf("Score[%d] = %d\n", 5, score[5]);
11     return 0;
12 }
13
```

C:\Users\Joke-ICT\Desktop\test1.exe

```
Input score 0 : 25
Input score 1 : 30
Input score 2 : 27
Input score 3 : 32
Input score 4 : 29
Score[0] = 25
Score[5] = 0
```

```
Process returned 0 (0x0)   execution time : 11.169 s
Press any key to continue.
```

การอ้างอิงสมาชิกใน **Array**

```
1  #include <stdio.h>
2  int main() {
3      int i, score[5];
4      for (i=0; i<=4; i++)
5      {
6          printf("Input score %d : ", i);
7          scanf("%d", &score[i]);
8      }
9      for (i=0; i<=4; i++)
10     {
11         printf("score[%d] : %d\n", i, score[i]);
12     }
13
14     return 0;
15 }
16
```

C:\Users\Joke-ICT\Desktop\test1.e

```
Input score 0 : 1
Input score 1 : 2
Input score 2 : 3
Input score 3 : 4
Input score 4 : 5
score[0] : 1
score[1] : 2
score[2] : 3
score[3] : 4
score[4] : 5
```

```
Process returned 0 (0x0)
Press any key to continue.
```

ตรวจสอบความเข้าใจกันหน่อย...

```
1  #include <stdio.h>
2  int main() {
3      int i, score[8];
4      for(i=0; i<=4; i++)
5      {
6          printf("Input score %d : ", i);
7          scanf("%d", &score[i]);
8      }
9      return 0;
10 }
11
```

```
Input score 0 : 20
Input score 1 : 15
Input score 2 : 10
Input score 3 : 25
Input score 4 : 35

Process returned 0 (0x0)
Press any key to continue.
```

จงบอกค่าของตัวแปรต่อไปนี้

1. score[1]
2. score[0] + score[4]
3. score[2+3]
4. score[2] + 10

มาทำโจทย์ง่าย ๆ กันนะ

Array01_1.c เขียนโปรแกรมรับค่าคะแนน (จำนวนเต็ม 0 - 100) ของนักเรียน 10 คน
เข้าไปเก็บในอาร์เรย์

Array01_2.c ปรับปรุงโค้ดคำสั่งข้อ Array01_1.c โดยให้แสดงคะแนนคนสุดท้ายย้อน
กลับไปคนที่ 1

Array01_3.c ปรับปรุงโค้ดคำสั่งข้อ Array01_1.c โดยให้หาผลรวมของคะแนนทั้งหมด
และค่าเฉลี่ยของคะแนนนักเรียนกลุ่มนี้

Array01_4.c ปรับปรุงโค้ดคำสั่งข้อ Array01_2.c ให้นับคนที่สอบผ่านและสอบตกว่ามี
กี่คน (ผ่านเกณฑ์ ≥ 50 คะแนน)

ขนาดของ Array

```
1  #include <stdio.h>
2  int main() {
3      int score[10];
4      float gpa[10];
5      char name[10];
6      printf("Size of score[10] = %d\n", sizeof(score));
7      printf("Size of gpa[10] = %d\n", sizeof(gpa));
8      printf("Size of name[10] = %d\n", sizeof(name));
9      return 0;
10 }
11
```

Select C:\Users\Joke-ICT\Desktop\test1.exe

```
Size of score[10] = 40
Size of gpa[10] = 40
Size of name[10] = 10
```

```
Process returned 0 (0x0)   execution time : 0.023 s
Press any key to continue.
```

การกำหนดค่าเริ่มต้นให้ **Array**

ชนิดของข้อมูล ชื่อตัวแปรอาร์เรย์[ขนาด] = {value-list};

ตัวอย่าง

- `int n[5] = {1, 4, 9, 16, 25};`
- `char a[3] = { 'A', 'B', 'C' };`
- `int pw[] = {1, 2, 4, 8, 16, 32, 64, 128};`
- `char name[] = "COMPUTER";`

↑
ถ้าไม่ระบุขนาด โปรแกรมจะจองหน่วยความจำให้เอง

การกำหนดค่าเริ่มต้นให้ **Array**

- ค่าคงที่ที่ใช้ในการกำหนดค่าเริ่มต้น จะต้องมีชนิดสอดคล้องกับชนิดอาร์เรย์ ค่าคงที่แต่ละค่าจะถูกนำไปกำหนดให้กับสมาชิกแต่ละตัวตามลำดับ
- หากเราใช้ค่าคงที่จำนวนน้อยกว่าสมาชิกของอาร์เรย์ ตำแหน่งที่เหลือจะถูกกำหนดให้เป็น 0

เช่น

```
int a[5] = {12 , 7 };
```

มีค่าเท่ากับ

```
int a[5] = {12 , 7 , 0 , 0 , 0 };
```

การกำหนดค่าเริ่มต้นให้ **Array**

การประกาศตัวแปรแบบอาร์เรย์พร้อมทั้งกำหนดค่าเริ่มต้น สามารถทำได้โดยไม่ต้องกำหนดขนาดของอาร์เรย์ก็ได้ คอมไพเลอร์จะหาจำนวนสมาชิกในกลุ่มค่าคงที่เอง เช่น

```
char codes[6] = {'s', 'a', 'm', 'p', 'l', 'e'};
```

```
char codes[] = {'s', 'a', 'm', 'p', 'l', 'e'};
```

ทั้งสองบรรทัดทำงานเหมือนกันโดยคอมไพเลอร์จะรู้ได้เองว่าขนาดของอาร์เรย์จะมีขนาดเป็น 6 แต่ห้ามเว้นไว้โดยไม่กำหนดอะไรเลย

เช่น

```
char codes[ ];
```

โจทย์ง่าย ๆ มาอีกแล้ว

Array02_1.c จงเขียนโปรแกรมเพื่อหาค่ามากที่สุดของอาร์เรย์ จากโปรแกรมต่อไปนี้

```
#include <stdio.h>
int main()
{
    int a[5] = {-1, 6, 9, 2, -9};
    //coding
    return 0;
}
```

Array02_2.c จากข้อ Array02_1.c ให้หาค่าน้อยที่สุดของอาร์เรย์

Array02_3.c จากข้อ Array02_2.c ให้หาค่าสมบูรณ์ของอาร์เรย์ a เก็บไว้ในอาร์เรย์ b

มาเขียนโปรแกรมสร้างกราฟกัน

Array03_1.c จงเขียนโปรแกรมรับค่าข้อมูล

จำนวนเต็ม (0-20) ไปเก็บในอาร์เรย์

แล้วให้แสดงผลดังนี้


C:\Users\Joke-ICT\Desktop\test1.exe

```
Input 0 : 9
Input 1 : 11
Input 2 : 15
Input 3 : 0
Input 4 : 2
Input 5 : 7
Input 6 : 4
Input 7 : 17
Input 8 : 9
Input 9 : 1
```

Element	value
0	9
1	11
2	15
3	0
4	2
5	7
6	4
7	17
8	9
9	1

มาเขียนโปรแกรมสร้างกราฟกัน

Array03_2.c ปรับปรุงโปรแกรมในข้อ Array03_1.c ให้แสดงผลดังนี้

 C:\Users\Joke-ICT\Desktop\test1.exe

```
Input 0 : 1
Input 1 : 2
Input 2 : 3
Input 3 : 10
Input 4 : 15
Input 5 : 19
Input 6 : 20
Input 7 : 5
Input 8 : 0
Input 9 : 7
Element      value      Histogram
    0          1          *
    1          2         **
    2          3        ***
    3         10       *****
    4         15      *****
    5         19     *****
    6         20     *****
    7          5      *****
    8          0
    9          7      *****
```

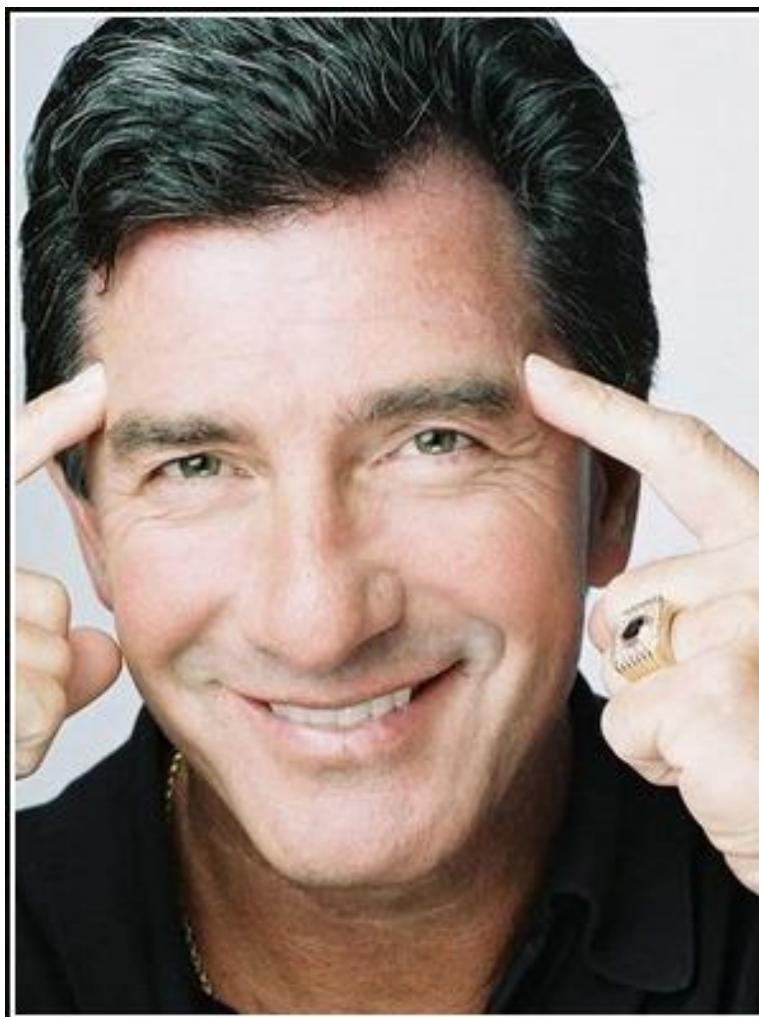
วิเคราะห์โจทย์...บอก Output

```
1  #include <stdio.h>
2  #include <math.h>
3  int main() {
4      int i,j,n[5]={-10,10,3,-9,3},m[5];
5      for(i=0;i<5;i++)
6      {
7          if(n[i]<0) m[4-i] = abs(n[i]);
8          else m[4-i] = pow(n[i],i);
9      }
10     for(i=0;i<5;i++)
11         printf("%d ",m[i]);
12     return 0;
13 }
```

C:\Users\Joke-ICT\Desktop\test1

81 9 9 10 10
Process returned 0 (0x0)

พักซักแปะปึง



What you hear, you forget; what you
see, you remember; what you do,
you understand.

— T. Harv Eker —

AZ QUOTES

พัคซัคแพปนัอง

Davinci Code

Array 2 มิติ



- การประกาศตัวแปรอาร์เรย์สองมิติจะใช้ดัชนี 2 ตัว เพื่อระบุจำนวนสมาชิกในแต่ละหลัก และแต่ละแถว ดังนี้

ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์[Row][Column]

Array 2 มิติ



ตัวอย่างเช่น

```
int AB[2][3];
```

จะมีสมาชิกทั้งหมด 6 ตัว (2×3) การอ้างสมาชิกแต่ละตัวทำได้ดังนี้

แถวที่ 0 $AB[0][0]$, $AB[0][1]$, $AB[0][2]$

แถวที่ 1 $AB[1][0]$, $AB[1][1]$, $AB[1][2]$

การกำหนดค่าเริ่มต้นให้ **Array** 2 มิติ

ตัวอย่าง

```
int    sqr[3][3]    =    {  
                                1,  2,  3,  
                                4,  5,  6,  
                                7,  8,  9  
                                };
```

```
int    B[2][2]      =    { {1, 2}, {3, 4} };
```

การกำหนดค่าเริ่มต้นให้ **Array 2 มิติ**

- หากไม่มีการกำหนดจำนวนแถว คอมไพเลอร์จะกำหนดจำนวนแถวโดยนับจากที่กำหนดในค่าเริ่มต้น แต่จะต้องมีการกำหนดจำนวนหลักเสมอ เช่น

```
int a[][2] = { {5, 8}, {9}, {-1} };
```

	0	1
0	5	8
1	9	0
2	-1	0

การกำหนดค่าเริ่มต้นให้ **Array** 2 มิติ

เช่น อาร์เรย์ที่มีสมาชิกอยู่ 3 ตัว และสมาชิกแต่ละตัวก็เป็นอาร์เรย์เก็บข้อมูลชนิด int มีจำนวนสมาชิก 2 ตัว จะกำหนดได้ดังนี้

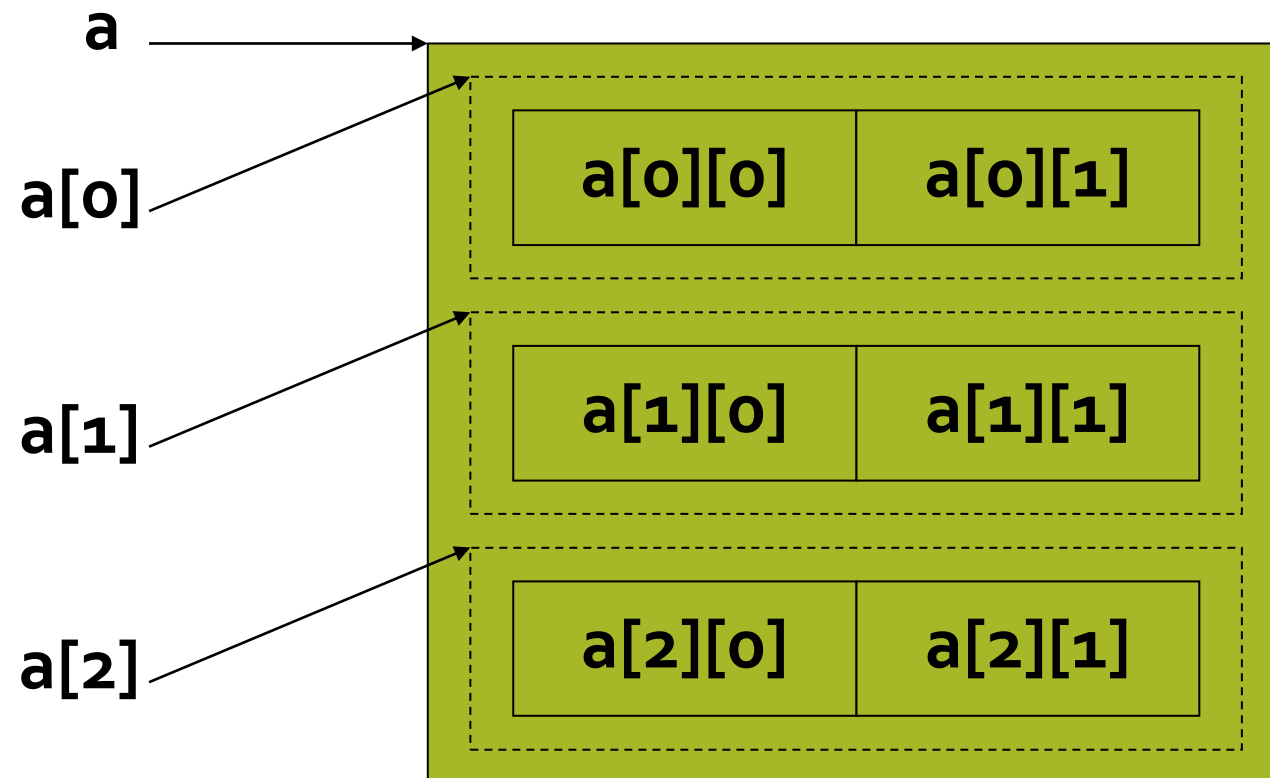
```
int a[3][2];
```

- ขนาดของอาร์เรย์ 3 x 2
- จำนวนไบต์ ที่ใช้ในการเก็บอาร์เรย์

sizeof (a) คำนวณจาก $3 \times 2 \times \text{sizeof}(\text{int}) = 3 \times 2 \times 4 = 24$ ไบต์

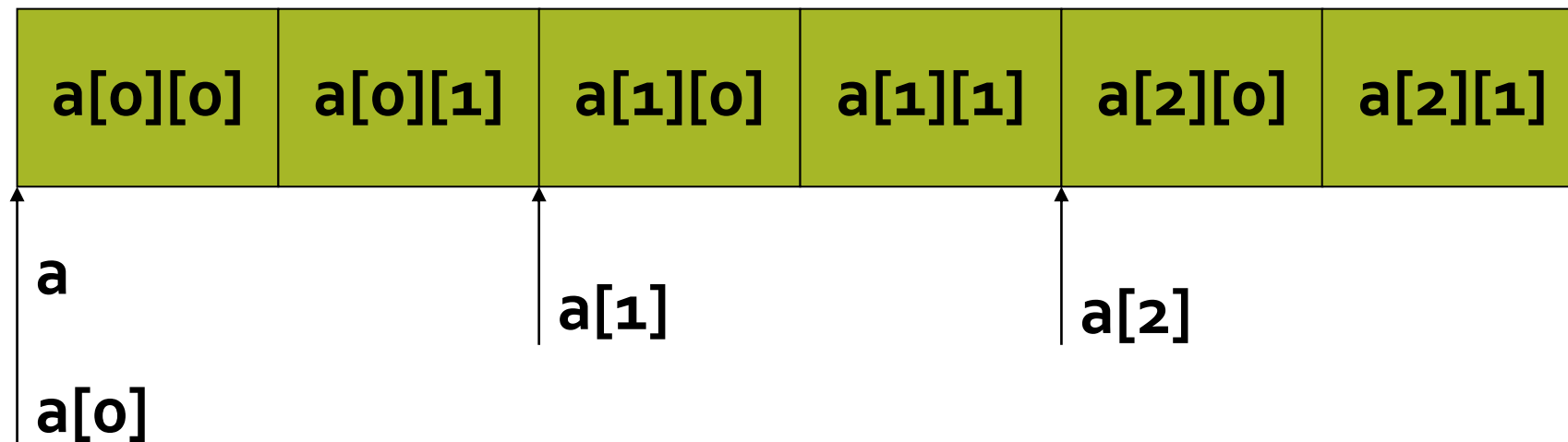
การกำหนดค่าเริ่มต้นให้ **Array** 2 มิติ

จากการกำหนดดังกล่าว จะได้อาร์เรย์ที่มีโครงสร้างดังรูป



การกำหนดค่าเริ่มต้นให้ **Array** 2 มิติ

การจัดวางภายในหน่วยความจำ สามารถแสดงได้ดังรูป



มาทดสอบกันหน่อย

```
int a[][2] = { {5, 8}, {9}, {-1} };
```

`a[0][1] = ?`

`a[2][1] = ?`

`a[1][0] = ?`

อาร์เรย์นี้...มีขนาดเท่าใด

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

```
int main()
```

```
{
```

```
    int x[2][5];
```

```
    char names[3][4];
```

```
    float nums[2][2];
```

```
    printf("Size of x = %d\n",sizeof(x));
```

```
    printf("Size of names = %d\n",sizeof(names));
```

```
    printf("Size of nums = %d\n",sizeof(nums));
```

```
    return 0;
```

```
}
```

Size of x = 40

Size of names = 12

Size of nums = 16

ตัวอย่าง การเก็บข้อมูลในอาร์เรย์

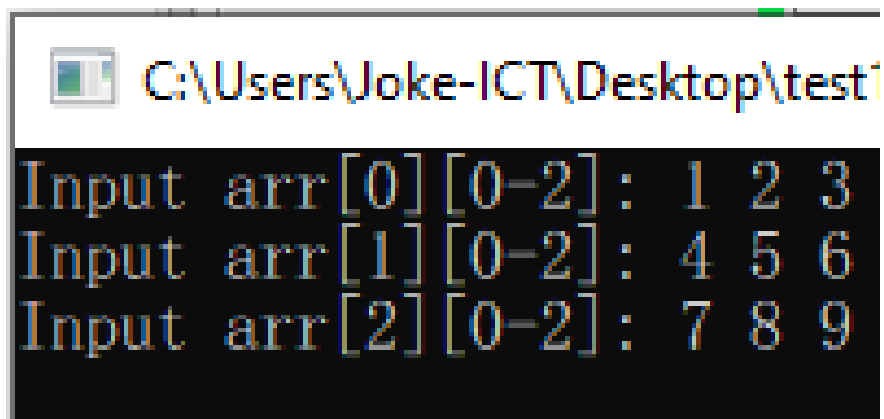
```
1  #include <stdio.h>
2  int main() {
3      int i,j,arr[3][3];
4      for(i=0;i<3;i++)
5      {
6          for(j=0;j<3;j++)
7          {
8              printf("Input arr[%d][%d]: ",i,j);
9              scanf("%d",&arr[i][j]);
10         }
11     }
12     return 0;
13 }
```

C:\Users\Joke-ICT\Desktop

```
Input arr[0][0]: 1
Input arr[0][1]: 2
Input arr[0][2]: 3
Input arr[1][0]: 4
Input arr[1][1]: 5
Input arr[1][2]: 6
Input arr[2][0]: 7
Input arr[2][1]: 8
Input arr[2][2]: 9
```

ตัวอย่าง การเก็บข้อมูลในอาร์เรย์

```
1  #include <stdio.h>
2  int main() {
3      int i,j,arr[3][3];
4      for(i=0;i<3;i++)
5      {
6          printf("Input arr[%d][0-2]: ",i);
7          scanf("%d %d %d",&arr[i][0],&arr[i][1],&arr[i][2]);
8      }
9      return 0;
10 }
```



```
C:\Users\Joke-ICT\Desktop\test
Input arr[0][0-2]: 1 2 3
Input arr[1][0-2]: 4 5 6
Input arr[2][0-2]: 7 8 9
```

ทำโจทย์กันดีกว่า

Array04_1.c เขียนโปรแกรมรับค่าคะแนน (จำนวนเต็ม 0 - 100) เก็บลงในอาร์เรย์ขนาด 3×3 แล้วแสดงผลให้อยู่ในรูปตาราง 3×3

Array04_2.c ปรับปรุงโปรแกรมข้อ Array04_1.c โดยนำค่าแต่ละ element ยกกำลังสองแล้วเก็บค่าไว้ในตัวแปรอาร์เรย์อีกตัวหนึ่ง แล้วทำการแสดงผลอาร์เรย์นั้นในรูปตาราง 3×3

ทำโจทย์กันดีกว่า

Array05_1.c สร้างเมตริกซ์ขนาด 3×2 จำนวน 2 เมตริกซ์ แล้วรับค่าจำนวนเต็มเก็บไว้ใน 2 เมตริกซ์นั้น จากนั้นทำการบวกเมตริกซ์ แล้วแสดงผลลัพธ์

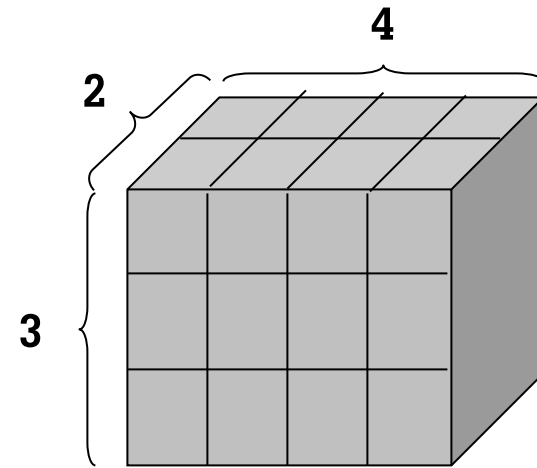
Array05_2.c สร้างเมตริกซ์ขนาด 3×2 เพื่อรับค่าจำนวนเต็ม แล้วแสดงผลลัพธ์เป็นทรานสโพสของเมตริกซ์นั้น

Array มากกว่า 2 มิติ

เช่น อาร์เรย์ 3 มิติ

```
int volume[3][4][2];  
int i,j,k;  
for (i=0; i<3; i++)  
    for (j=0; j<4; j++)  
        for (k=0; k<2; k++)  
            val[i][j][k] = 0;
```

- อีลีเมนต์แรกคือ `val[0][0][0]`
- อีลีเมนต์สุดท้าย `val[2][3][1]`



ความผิดพลาดของโปรแกรมทั่วไปเกี่ยวกับอาร์เรย์

- ลืมประกาศตัวแปร
- ใช้งานตัวแปรอาร์เรย์ นอกเหนือจากขอบเขตที่ประกาศไว้

```
int a_error[3] , i;
```

```
for ( i = 0 ; i <= 3 ; i++ )
```

```
    a_error[i] = i;
```

ความผิดพลาดของโปรแกรมทั่วไปเกี่ยวกับอาร์เรย์

- ลืมให้ค่าเริ่มต้นกับตัวแปรอาร์เรย์ แล้วนำตัวแปรนั้นไปใช้งาน

```
int a1 = 5, a2 = 8;
```

```
int total[2];
```

```
total[1] = total[0] + a1 + a2;
```

การให้ค่าเริ่มต้นอย่างง่าย `int array[10][5][3]={ };` ทุก element เป็นศูนย์

- กำหนดค่าให้กับตัวแปรอาร์เรย์โดยตรง

```
int x[10];
```

```
x = 11;
```


ข้อควรระวังการใช้งานอาร์เรย์

- เมื่อประกาศตัวแปรอาร์เรย์และกำหนดขนาดของอาร์เรย์แล้ว เราอาจจะเขียนหรืออ่านเกินกว่าสมาชิกของอาร์เรย์ได้ โดยคอมไพเลอร์จะไม่แจ้งเตือนแต่อย่างใด
- แต่ การเขียนข้อมูลเกินกว่าขนาดของอาร์เรย์ที่จองไว้ จะทำให้ข้อมูลที่เขียนลงไปนั้น จะไปทับข้อมูลอื่น หรือไปทับส่วนของโปรแกรม อันจะทำให้ระบบเสียหายได้

The End

