

แถวลำดับ (Array)

อ.อภิเชก หงษ์วิทยากร
ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยศิลปากร
apisake@su.ac.th

ทบทวนกันหน่อย

```
char    letter;  
int     i_number;  
float   f_number;  
double  d_number;
```

ทบทวนกันหน่อย

```
// intro.c -- nothing but an introduction code

#include <stdio.h>

int main(void)
{
    int i;

    i=1;
    i=i+1;

    printf("%d\n",i);
}
```

ถ้าเราต้องใช้ตัวแปรหลายตัว?

```
// vars.c -- using multiple variables
#include <stdio.h>

int main(void)
{
    int m1,m2,m3,m4,m5;

    m1=1;
    m2=m1+1;
    m3=m2+1;
    m4=m3+1;
    m5=m4+1;

    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);
}
```

ถ้าเราต้องใช้ตัวแปรหลายตัว?

```
// vars.c -- using multiple variables  
#include <stdio.h>
```

```
int main(void)  
{  
    int m1,m2,m3,m4,m5;
```

1	2	3	4	5
---	---	---	---	---

```
    m1=1;  
    m2=m1+1;  
    m3=m2+1;  
    m4=m3+1;  
    m5=m4+1;
```

```
    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);  
}
```

มาสร้างตารางสูตรคูณกัน...

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

โปรแกรมแสดงตารางสูตรคูณ

```
// mult_vars.c -- multiplication table using multiple variables
#include <stdio.h>

int main(void)
{
    int m1,m2,m3,m4,m5;

    m1=1;
    m2=m1+1;
    m3=m2+1;
    m4=m3+1;
    m5=m4+1;

    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);

}
```

โปรแกรมแสดงตารางสูตรคูณ

```
// mult_vars.c -- multiplication table using multiple variables
#include <stdio.h>

int main(void)
{
    int m1,m2,m3,m4,m5;

    m1=1;
    m2=m1+1;
    m3=m2+1;
    m4=m3+1;
    m5=m4+1;

    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);
    printf("%5d %5d %5d %5d %5d\n",m1*2,m2*2,m3*2,m4*2,m5*2);
}
```

โปรแกรมแสดงตารางสูตรคูณ

```
// mult_vars.c -- multiplication table using multiple variables  
#include <stdio.h>
```

```
int main(void)  
{  
    int m1,m2,m3,m4,m5;  
  
    m1=1;  
    m2=m1+1;  
    m3=m2+1;  
    m4=m3+1;  
    m5=m4+1;  
  
    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);  
    printf("%5d %5d %5d %5d %5d\n",m1*2,m2*2,m3*2,m4*2,m5*2);  
}
```

1	2	3	4	5
2	4	6	8	10

โปรแกรมแสดงตารางสูตรคูณ

```
// mult_vars.c -- multiplication table using multiple variables  
#include <stdio.h>
```

```
int main(void)  
{  
    int m1,m2,m3,m4,m5;  
  
    m1=1;  
    m2=m1+1;  
    m3=m2+1;  
    m4=m3+1;  
    m5=m4+1;  
  
    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);  
    printf("%5d %5d %5d %5d %5d\n",m1*2,m2*2,m3*2,m4*2,m5*2);  
}
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

โปรแกรมแสดงตารางสูตรคูณ

```
// mult_vars.c -- multiplication table using multiple variables
#include <stdio.h>
```

```
int main(void)
{
    int m1,m2,m3,m4,m5;
```

```
    m1=1;
    m2=m1+1;
    m3=m2+1;
    m4=m3+1;
    m5=m4+1;
```

```
    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);
    printf("%5d %5d %5d %5d %5d\n",m1*2,m2*2,m3*2,m4*2,m5*2);
    printf("%5d %5d %5d %5d %5d\n",m1*3,m2*3,m3*3,m4*3,m5*3);
    printf("%5d %5d %5d %5d %5d\n",m1*4,m2*4,m3*4,m4*4,m5*4);
    printf("%5d %5d %5d %5d %5d\n",m1*5,m2*5,m3*5,m4*5,m5*5);
```

```
}
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

ขยายตาราง...

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

ขยายตาราง...

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

```
// mult_vars.c -- multiplication table using multiple variables
#include <stdio.h>

int main(void)
{
    int m1,m2,m3,m4,m5;

    m1=1;
    m2=m1+1;
    m3=m2+1;
    m4=m3+1;
    m5=m4+1;

    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);
    printf("%5d %5d %5d %5d %5d\n",m1*2,m2*2,m3*2,m4*2,m5*2);
    printf("%5d %5d %5d %5d %5d\n",m1*3,m2*3,m3*3,m4*3,m5*3);
    printf("%5d %5d %5d %5d %5d\n",m1*4,m2*4,m3*4,m4*4,m5*4);
    printf("%5d %5d %5d %5d %5d\n",m1*5,m2*5,m3*5,m4*5,m5*5);
}
```

ขยายตาราง...

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

```
// mult_vars.c -- multiplication table using multiple variables
#include <stdio.h>

int main(void)
{
    int m1,m2,m3,m4,m5;

    m1=1;
    m2=m1+1;
    m3=m2+1;
    m4=m3+1;
    m5=m4+1;

    printf("%5d %5d %5d %5d %5d\n",m1,m2,m3,m4,m5);
    printf("%5d %5d %5d %5d %5d\n",m1*2,m2*2,m3*2,m4*2,m5*2);
    printf("%5d %5d %5d %5d %5d\n",m1*3,m2*3,m3*3,m4*3,m5*3);
    printf("%5d %5d %5d %5d %5d\n",m1*4,m2*4,m3*4,m4*4,m5*4);
    printf("%5d %5d %5d %5d %5d\n",m1*5,m2*5,m3*5,m4*5,m5*5);
}
```



ทำอย่างไรดี...?



ทำอย่างไรดี...?

ใช้ array ช่วย

Array คืออะไร

Array คืออะไร

int
double
float
char

Array คืออะไร



Array คืออะไร

Array คืออะไร

int m1



Array คืออะไร

int m1

int m2

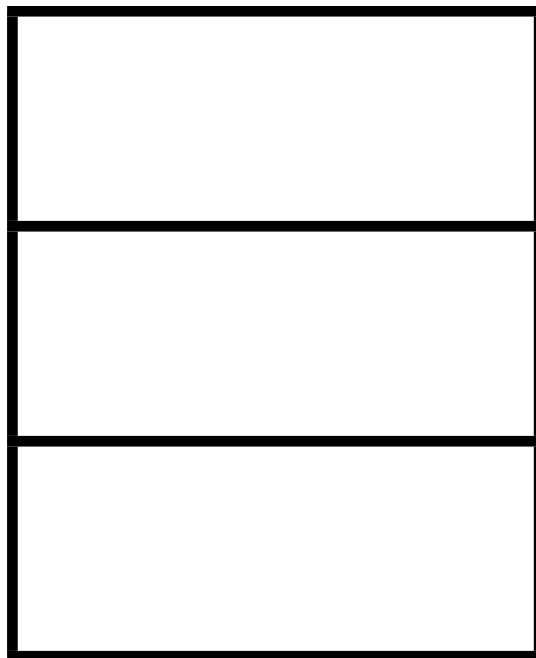


Array คืออะไร

int m1

int m2

int m3



Array คืออะไร

int m1

int m2

int m3

int m4



Array คืออะไร

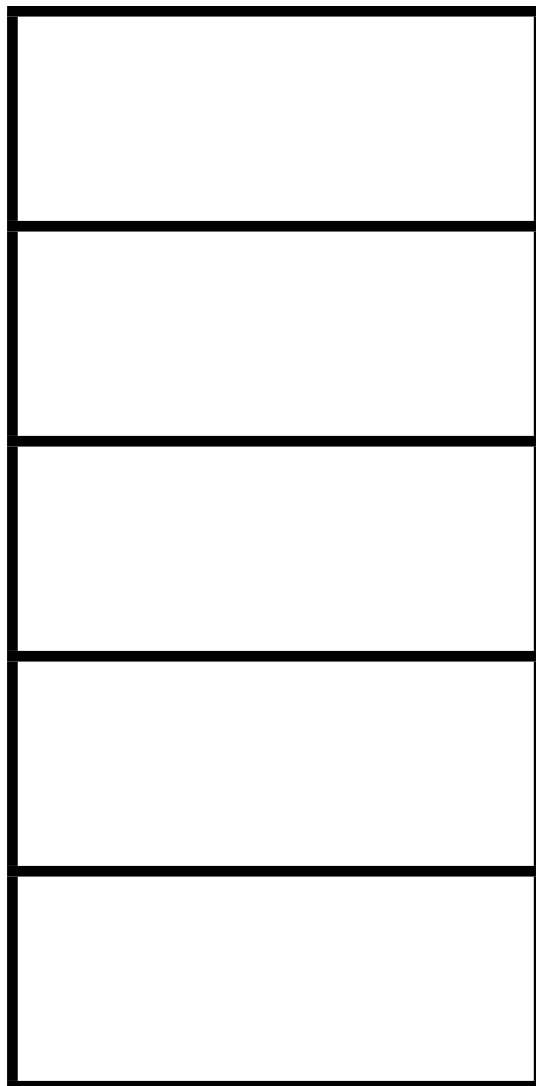
int m1

int m2

int m3

int m4

int m5



Array คืออะไร

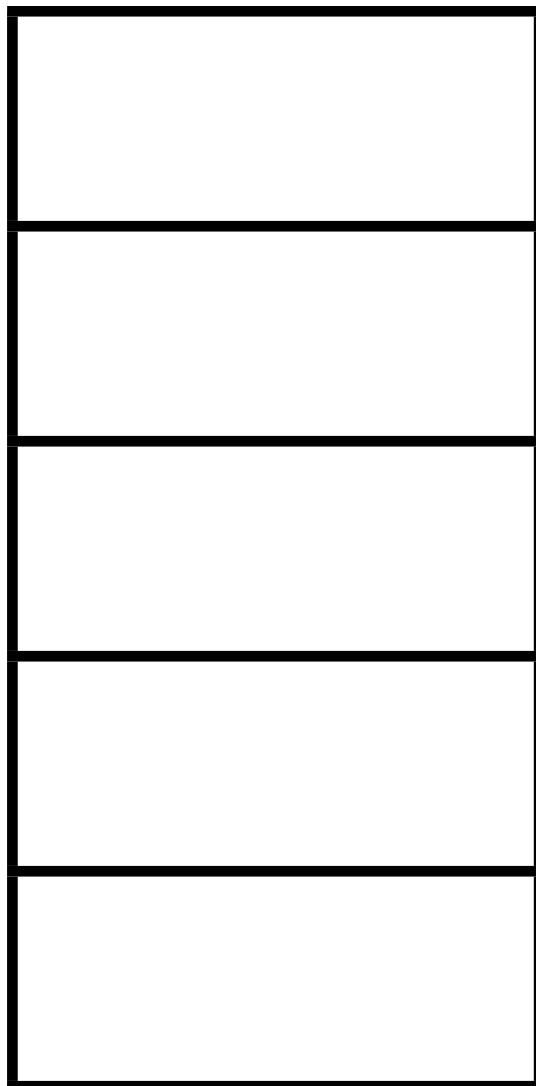
int m1

int m2

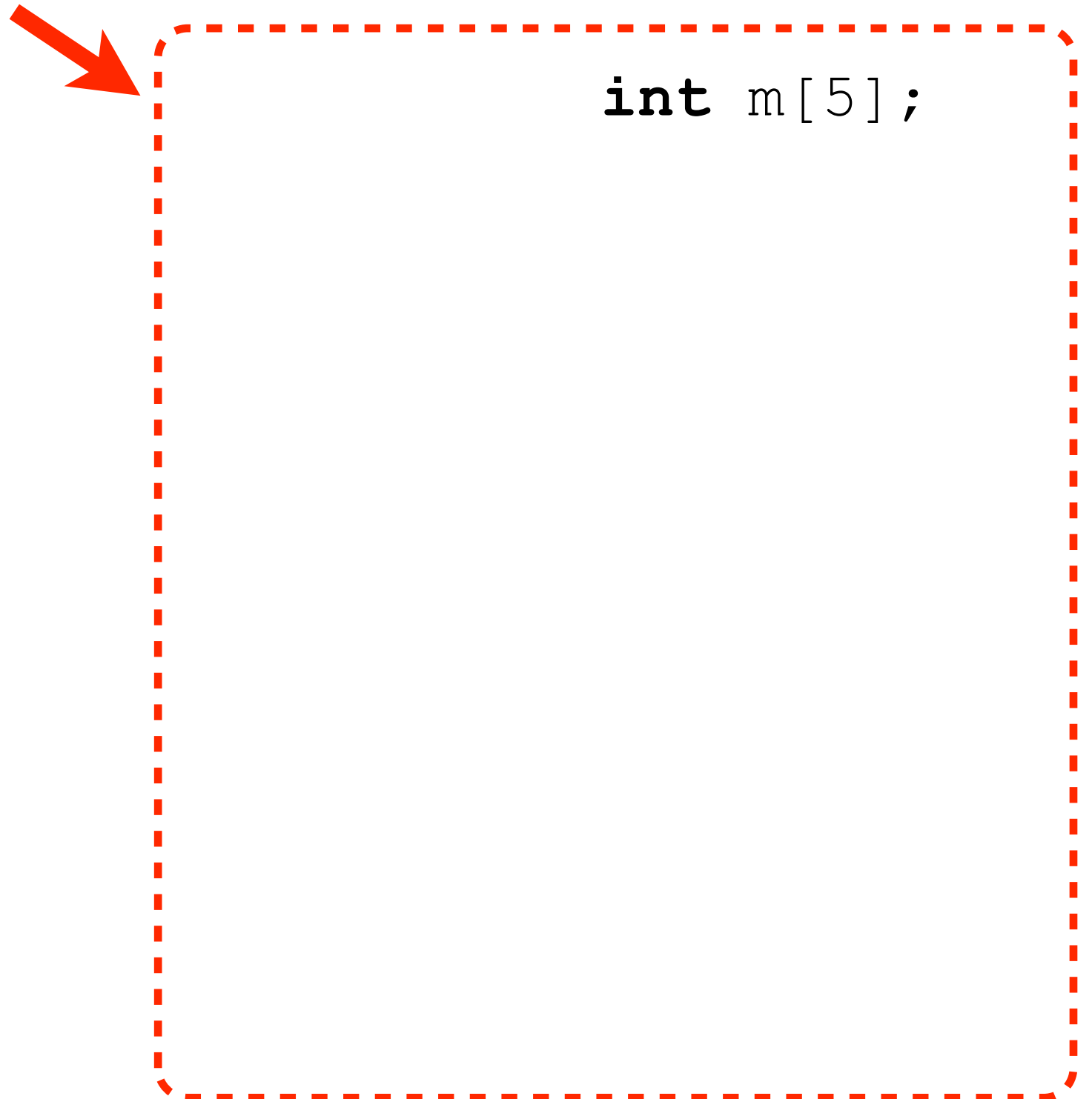
int m3

int m4

int m5



int m[5];



Array คืออะไร

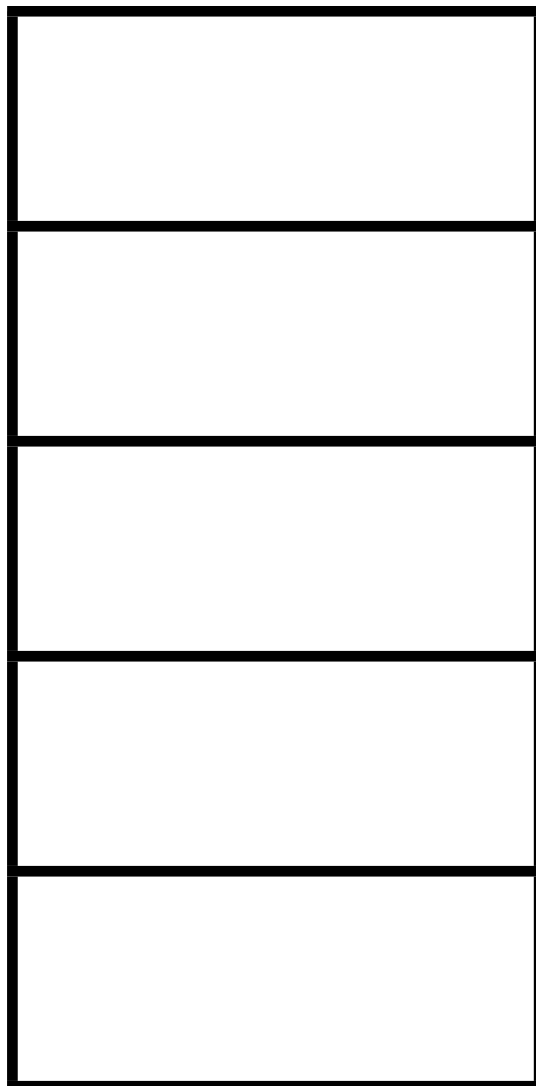
int m1

int m2

int m3

int m4

int m5



m[0]

int m[5];



Array คืออะไร

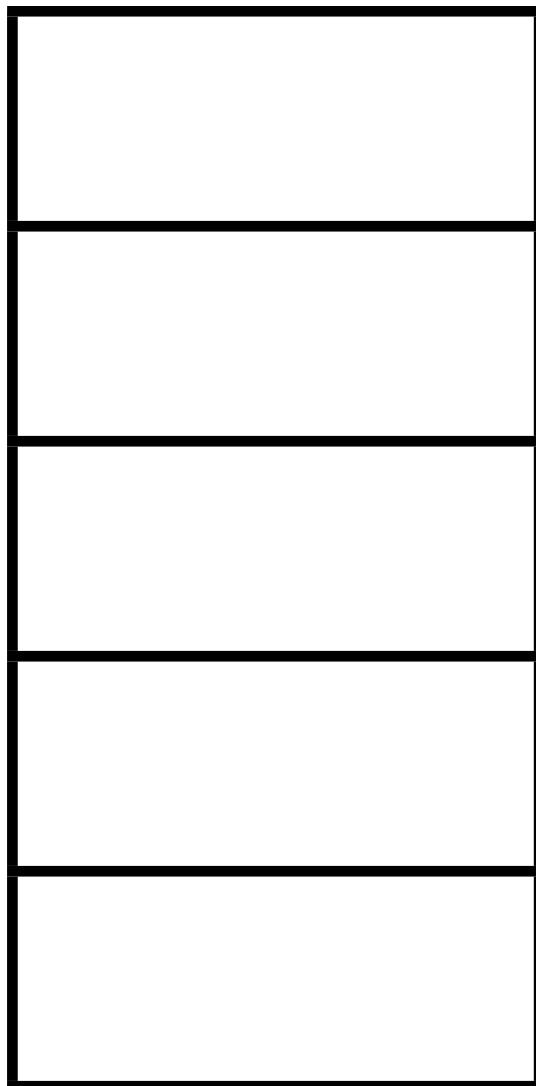
int m1

int m2

int m3

int m4

int m5



m[0]

m[1]

int m[5];



Array คืออะไร

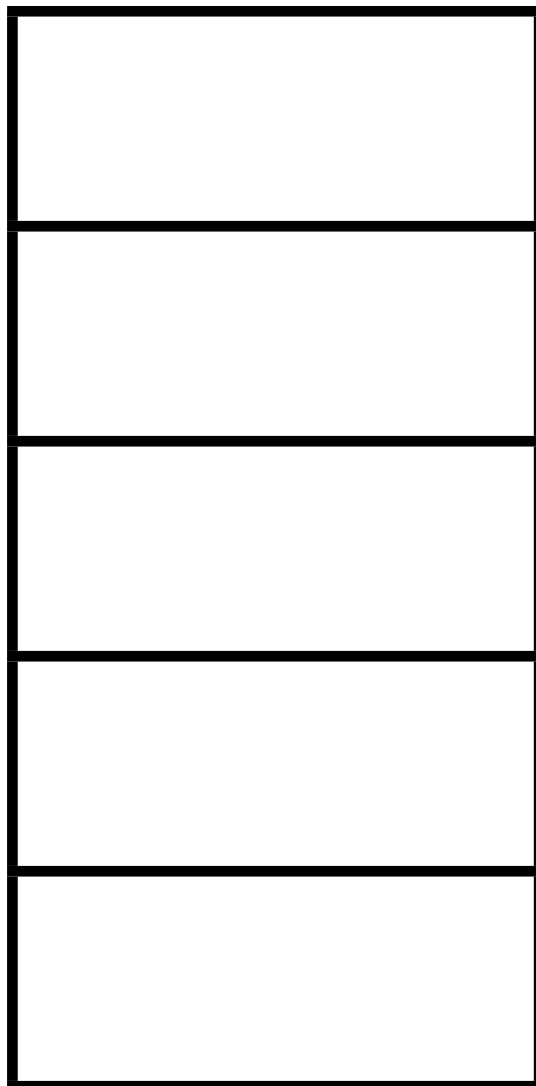
int m1

int m2

int m3

int m4

int m5

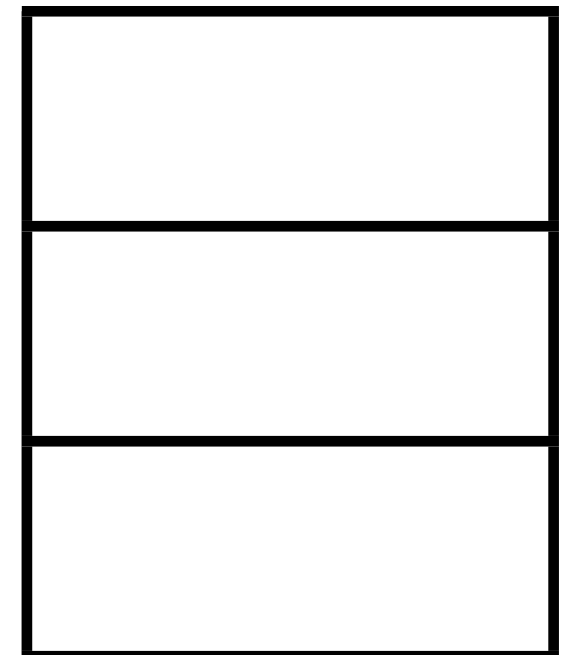


m[0]

m[1]

m[2]

int m[5];



Array คืออะไร

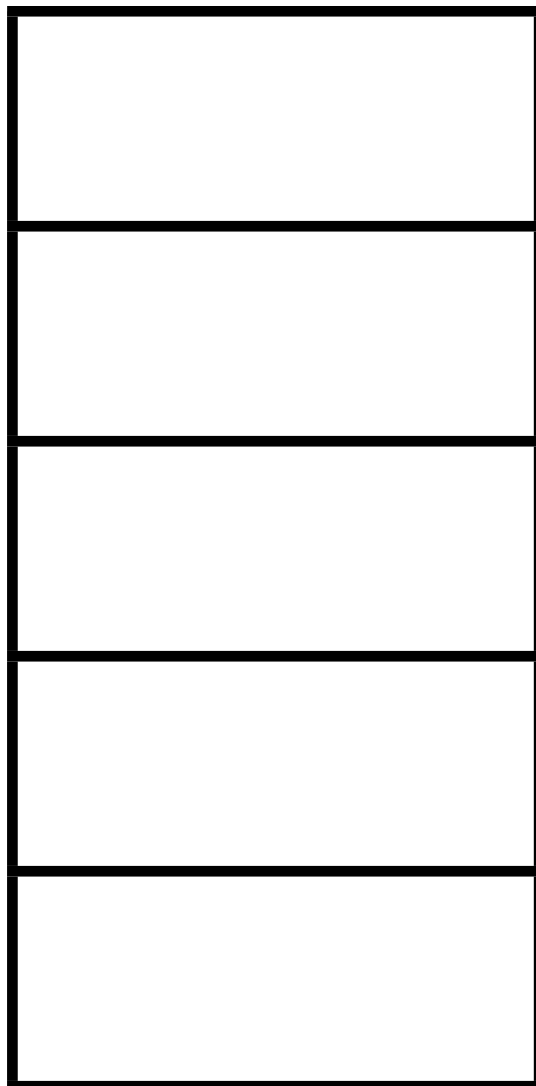
int m1

int m2

int m3

int m4

int m5



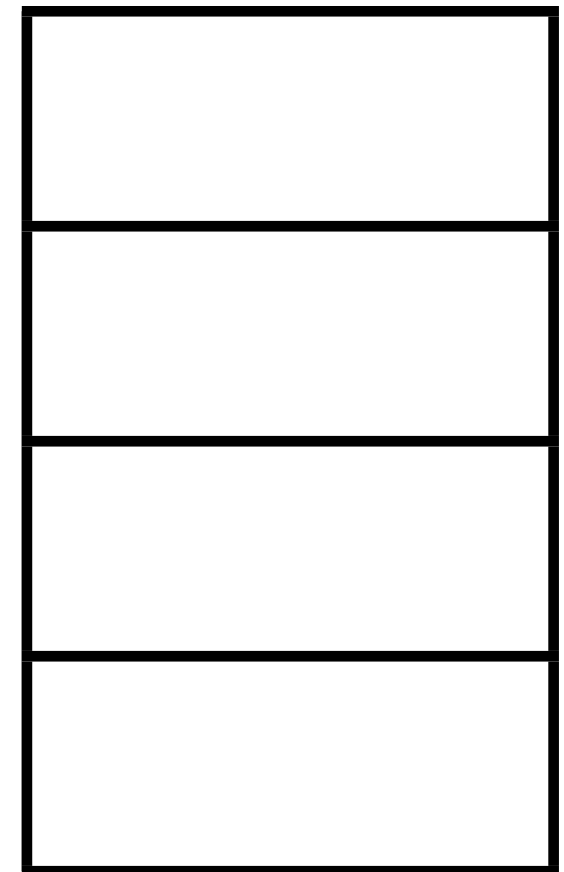
int m[5];

m[0]

m[1]

m[2]

m[3]



Array คืออะไร

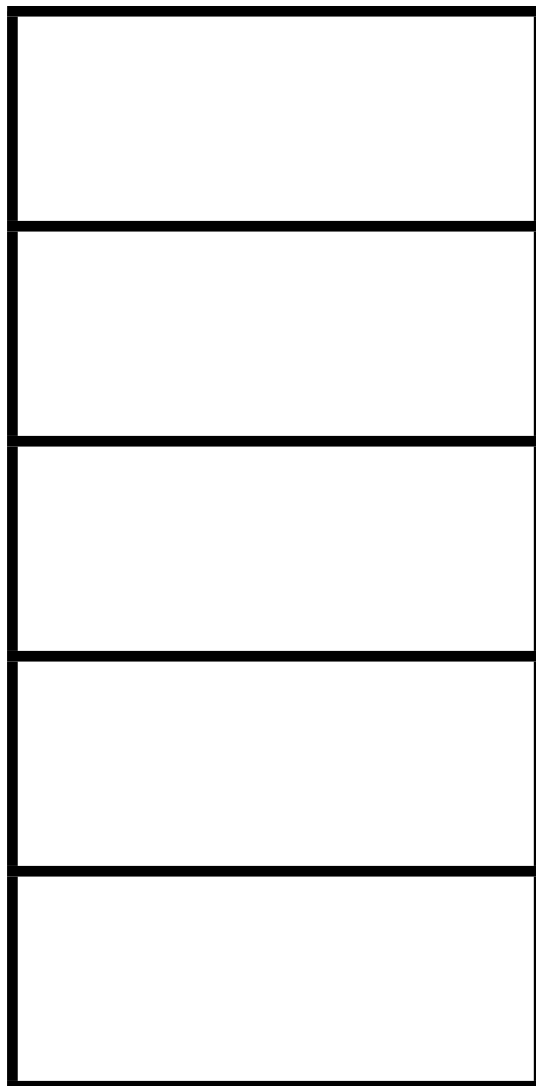
int m1

int m2

int m3

int m4

int m5



int m[5];

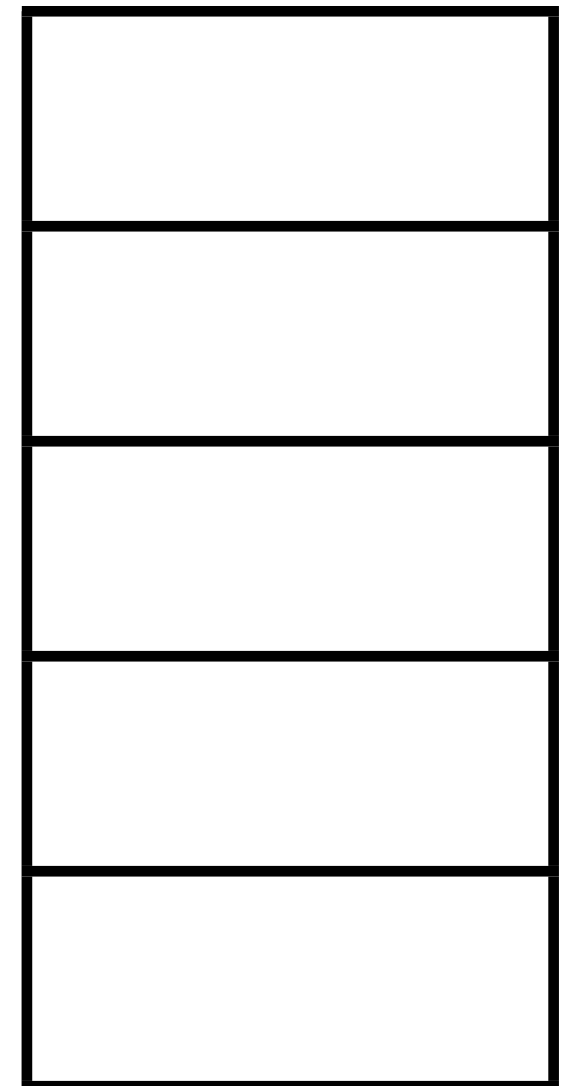
m[0]

m[1]

m[2]

m[3]

m[4]



การใช้ Array 1 มิติ

การประกาศตัวแปร:

data-type *array[expression];*

เช่น

```
char    name[20];  
int     i[5];  
float   f[10];  
double  d[7];
```

การใช้ Array 1 มิติ

การเรียกใช้ค่าจาก array:

เรียกใช้โดยอ้างอิงเลขดัชนีของ array เริ่มจาก
0 จนถึง n-1

เช่น

```
int m[5];
```

```
    .    .  
    .    .
```

```
m[0]=1;
```

```
m[1]=2;
```

```
m[2]=3;
```

```
m[3]=4;
```

```
m[4]=5;
```

```
// array_demo01.c -- demo on array step1
```

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

```
int main(void)
```

```
{
```

```
    int m[5];
```

```
    m[0]=1;
```

```
    m[1]=2;
```

```
    m[2]=3;
```

```
    m[3]=4;
```

```
    m[4]=5;
```

```
    printf("%5d %5d %5d %5d %5d\n",m[0],m[1],m[2],m[3],m[4]);  
}
```

การใช้ Array 1 มิติ

การกำหนดค่าตั้งต้น:

```
int array[n]={value_0,value_1,...,value_{n-1}};
```

เช่น

```
int m[5]={1,2,3,4,5};
```

```
// array_demo02.c -- demo on array step2
#include <stdio.h>

int main(void)
{
    int m[5]={1,2,3,4,5};

    printf("%5d %5d %5d %5d %5d\n",m[0],m[1],m[2],m[3],m[4]);
}
```

การใช้ Array 1 มิติ

ดรรชนีของ array มีประโยชน์ในการอ้างอิง โดยใช้ loop

```
// array_demo03.c -- demo on array step3
#include <stdio.h>

int main(void)
{
    int m[5];
    int i,j;

    for (i=0; i<5; i++) {
        m[i]=i+1;
    }

    for (j=0; j<5; j++) {
        printf("%5d",m[j]);
    }

    printf("\n");
}
```

การใช้ Array 1 มิติ

ดัชนีของ array มีประโยชน์ในการอ้างอิง โดยใช้ loop

```
// array_demo03.c -- demo on array step3
#include <stdio.h>
```

```
int main(void)
{
    int m[5];
    int i,j;

    for (i=0; i<5; i++) {
        m[i]=i+1;
    }

    for (j=0; j<5; j++) {
        printf("%5d",m[j]);
    }

    printf("\n");
}
```

1	2	3	4	5
---	---	---	---	---

การใช้ Array 1 มิติ

ขยายผลให้ได้ 5x5

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

```
// array_demo04.c -- demo on array step4
#include <stdio.h>

int main(void)
{
    int m[5];
    int i,j,k;

    for (i=0; i<5; i++) {
        m[i]=i+1;
    }

    for (j=0; j<5; j++) {
        for (k=0; k<5; k++) {
            printf("%5d",m[j]*(k+1));
        }
        printf("\n");
    }
}
```

การใช้ Array 1 มิติ

กลับมาที่ตารางผลลัพธ์ที่ต้องการ

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

```
// array_demo05.c -- demo on array step5
#include <stdio.h>
#define MAX 12

int main(void)
{
    int m[MAX];
    int i,j,k;

    for (i=0; i<MAX; i++) {
        m[i]=i+1;
    }

    for (j=0; j<MAX; j++) {
        for (k=0; k<MAX; k++) {
            printf("%5d",m[j]*(k+1));
        }
        printf("\n");
    }
}
```

การใช้ Array 2 มิติ

การประกาศตัวแปร:

data-type *array*[*expression 1*] [*expression 2*];

เช่น

int mat[3][2];

row 0

row 1

row 2

col 0

col 1

mat[0][0]	mat[0][1]
mat[1][0]	mat[1][1]
mat[2][0]	mat[2][1]

การใช้ Array 2 มิติ

การเก็บข้อมูลในหน่วยความจำ

คอมพิวเตอร์ทำการเก็บข้อมูลใน
ลักษณะ row-major order

mat[0][0]	mat[0][1]
mat[1][0]	mat[1][1]
mat[2][0]	mat[2][1]

การใช้ Array 2 มิติ

การเก็บข้อมูลในหน่วยความจำ

คอมพิวเตอร์ทำการเก็บข้อมูลใน
ลักษณะ row-major order

mat[0][0]	mat[0][1]
mat[1][0]	mat[1][1]
mat[2][0]	mat[2][1]

mat[0][0]	mat[0][1]	mat[1][0]	mat[1][1]	mat[2][0]	mat[2][1]

การใช้ Array 2 มิติ

การเก็บข้อมูลในหน่วยความจำ

คอมพิวเตอร์ทำการเก็บข้อมูลใน
ลักษณะ row-major order

mat[0][0]	mat[0][1]
mat[1][0]	mat[1][1]
mat[2][0]	mat[2][1]

mat[0][0]	mat[0][1]	mat[1][0]	mat[1][1]	mat[2][0]	mat[2][1]



การใช้ Array 2 มิติ

กล่าวคือ

เราสามารถกำหนดค่าเริ่มต้นได้ดังต่อไปนี้

```
int mat[3][2] = {1, 2, 3, 4, 5, 6};
```

mat[2][0]	mat[2][1]
5	6

** รายละเอียดอื่นๆให้อ่านได้ตั้งแต่วันที่ 23 ของเอกสารประกอบการเรียนฯ

การใช้ Array 2 มิติ

กล่าวคือ

เราสามารถกำหนดค่าเริ่มต้นได้ดังต่อไปนี้

```
int mat[3][2] = {1, 2, 3, 4, 5, 6};
```

mat[0][0]	mat[0][1]	mat[1][0]	mat[1][1]	mat[2][0]	mat[2][1]
1	2	3	4	5	6

** รายละเอียดอื่นๆให้อ่านได้ตั้งแต่วันที่ 23 ของเอกสารประกอบการเรียนฯ

การใช้ Array 2 มิติ

```
// 2d_array01.c -- demo on 2D array step1
#include <stdio.h>

int main(void)
{
    int score[4][3]=
    {
        {32,21,19},
        {20,18,25},
        {35,24,23},
        {28,15,16}
    };
    int i,j;

    for (i=0; i<4; i++) {
        for (j=0; j<3; j++) {
            printf("%5d",score[i][j]);
        }
        printf("\n");
    }
}
```

การใช้ Array 2 มิติ

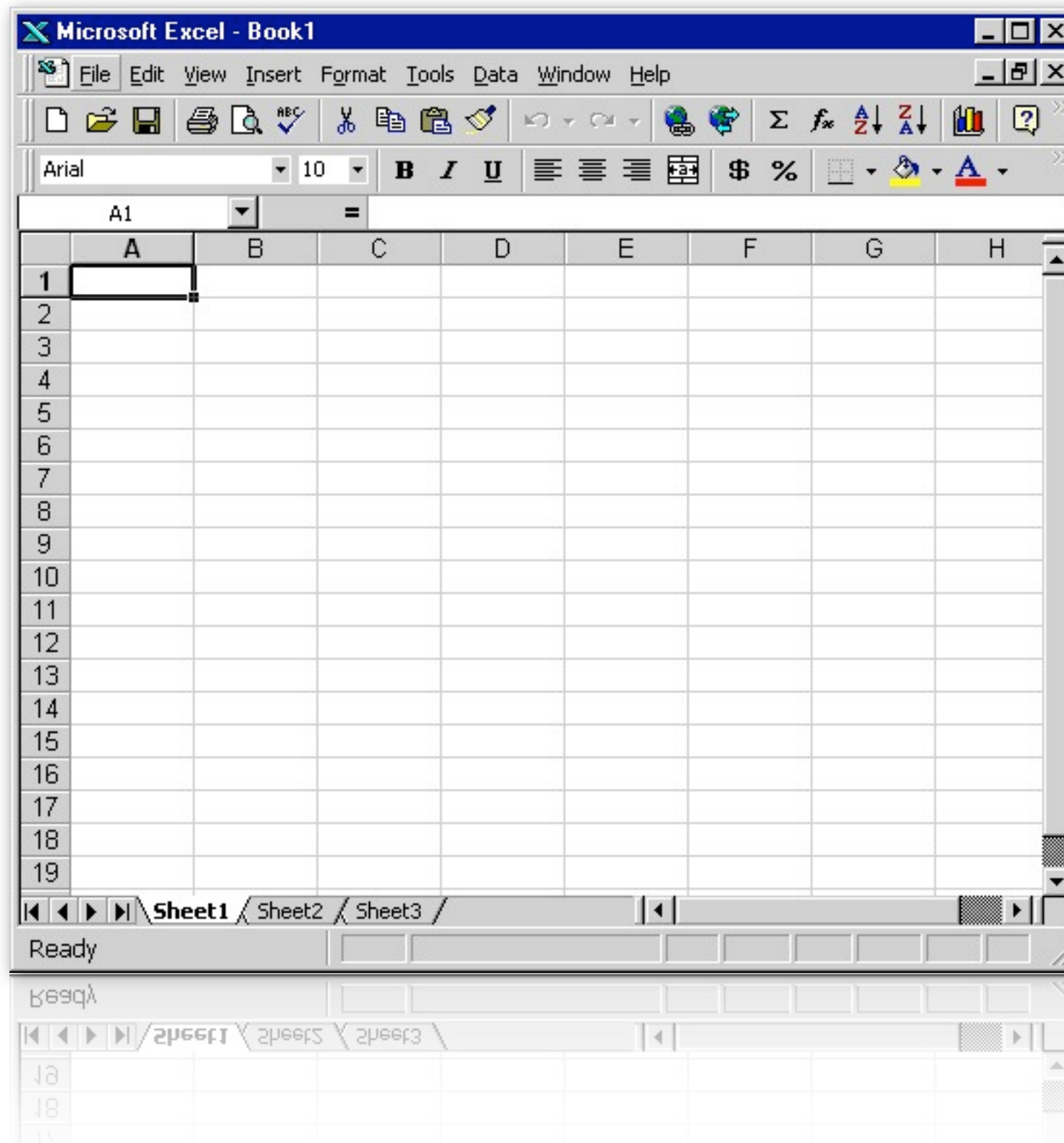
```
// 2d_array02.c -- demo on 2D array step2 (read user input)
#include <stdio.h>

int main(void)
{
    int score[4][3];
    int i,j;

    for (i=0; i<4; i++) {
        for (j=0; j<3; j++) {
            scanf("%d",&score[i][j]);
        }
    }

    for (i=0; i<4; i++) {
        for (j=0; j<3; j++) {
            printf("%5d",score[i][j]);
        }
        printf("\n");
    }
}
```

เราอาจนึกเปรียบเทียบ 2D array กับ Excel



ประกาศ

