

```
//38 address
```

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

```
int main()
```

```
{
```

```
    int con, a[10], b[10][10], c[10][10][10], *p = NULL, i, j, k, s1, s2, s3, counter, choice, address;
```

```
    con = 1;
```

```
    while(con == 1)
```

```
    {
```

```
        counter = 0;
```

```
        printf("choose an array (1 for 1D, 2 for 2D, 3 for 3D) : ");
```

```
        scanf("%d", &choice);
```

```
        //ให้เลือกมิติของ array
```

```
        if(choice == 1)
```

```
        //ถ้าเลือก 1 มิติ
```

```
        {
```

```
            printf("input first slot size (not more than 10) : ");
```

```
            scanf("%d", &s1);
```

```
            //ให้กำหนดขนาดของ array 1 มิติ โดยแต่ละช่องมีขนาดไม่เกิน 10
```

```
            if(s1 > 0 && s1 < 10)
```

```
            //ถ้าแต่ละช่องไม่เกิน 10
```

```
            {
```

```
                p = &a[0];
```

```
                //ให้ pointer p ใช้ไปที่ array a ช่องที่ 0 ที่เป็นจุดเริ่ม
```

```
                for(i = 0 ; i < s1 ; i++)
```

```
                {
```

```
                    a[i] = 0;
```

```
                    //set default
```

```
                    address = p + counter;
```

```
                    //ให้ address คือตำแหน่งของ p + counter
```

```
                    printf("address of a[%d] is %u\n", i, address);
```

```
                    //แสดงผลว่า ตำแหน่งของ array ช่องนั้นๆอยู่ที่ตำแหน่งที่เท่าไร
```

```
                    counter++;
```

```
                }
```

```
            }
```

```
            else
```

```
            {
```

```
                printf("wrong input\n");
```

```
            }
```

```
        }
```

```
else if(choice == 2)
```

```
//ถ้าเลือก 2 มิติ
```

```
{
    printf("input first slot size (not more than 10) : ");
    scanf("%d", &s1);
    printf("input secound slot size (not more than 10) : ");
    scanf("%d", &s2);
    //ให้กำหนดขนาดของ array 2 มิติ โดยแต่ละช่องมีขนาดไม่เกิน 10
    if((s1 > 0 && s1 < 10) && (s2 > 0 && s2 < 10))
        //ถ้าแต่ละช่องไม่เกิน 10
        {
            p = &b[0][0];
            //ให้ pointer p ใช้ไปที่ array b ช่องที่ 0 0 ที่เริ่มจุดเริ่ม
            printf("\n");
            for(i = 0 ; i < s1 ; i++)
            {
                for(j = 0 ; j < s2 ; j++)
                {
                    b[i][j] = 0;
                    //set defalut
                    address = p + counter;
                    //ให้ address คือตำแหน่งของ p + counter
                    printf("address of b[%d][%d] is %u\t", i, j, address);
                    //แสดงผลว่า ตำแหน่งของ array ช่องนั้นๆอยู่ที่ตำแหน่งที่เท่าไร
                    counter++;
                }
                printf("\n");
            }
            printf("\n");
        }
    else
    {
        printf("wrong input\n");
    }
}
```

```
else if (choice == 3)
```

```
//ถ้าเลือก 3 มิติ
```

```
{
    printf("input first slot size (not more than 10) : ");
    scanf("%d", &s1);
```

```

printf("input second slot size (not more than 10) : ");
scanf("%d", &s2);

printf("input third slot size (not more than 10) : ");
scanf("%d", &s3);

//ให้กำหนดขนาดของ array 3 มิติ โดยแต่ละช่องมีขนาดไม่เกิน 10
if((s1 > 0 && s1 < 10) && (s2 > 0 && s2 < 10) && (s3 > 0 && s3 < 10))
//ถ้าแต่ละช่องไม่เกิน 10
{
    p = &c[0][0][0];
    //ให้ pointer p ใช้ไปที่ array c ช่องที่ 0 0 0 ที่เป็นจุดเริ่ม
    for(k = 0 ; k < s3 ; k++)
    {
        printf("in k layer(s) %d \n", k);
        for(i = 0 ; i < s1 ; i++)
        {
            for(j = 0 ; j < s2 ; j++)
            {
                c[i][j][k] = 0;

                //set default
                address = p + counter;
                //ให้ address คือตำแหน่งของ p + counter
                printf("address of c[%d][%d][%d] is %u\t", i, j, k, address);
                //แสดงผลว่า ตำแหน่งของ array ช่องนั้นๆอยู่ที่ตำแหน่งที่เท่าไร
                counter++;
            }
            //อธิบาย counter ที่ใช้เป็น counter เนื่องจากว่าการใช้ (p + i + j + k) นั้นจะ

```

ตรงในช่วงแรกเท่านั้น คือช่วงที่ i, j เป็น 0 และ k มีค่าตั้งแต่ 0 - 2 เนื่องจาก

EX

ช่องที่	ค่า
0	1
1	2
2	3
3	4

การใช้ (p + i + j + k)

แล้ว i, j = 0 และ k = 0 จะได้ช่องที่ 0 ซึ่งตรงตามต้องการ

i, j = 0 และ k = 1 จะได้ช่องที่ 1 ซึ่งตรงตามต้องการ

i, j = 0 และ k = 2 จะได้ช่องที่ 2 ซึ่งตรงตามต้องการ

แต่ถ้าเป็น $i = 0, j = 1$ และ $k = 0$ จะได้ช่องที่ 1 ที่มีค่าเป็น 0 แต่สิ่งที่ต้องการ

คือช่องที่ 3 ที่มีค่าเป็น 4 จึงใช้ตัวแปร **counter** มาเพิ่มค่าทีละ 1 แทนการใช้ $i + j + k$

```
        */
    }
    printf("\n");
}
printf("\n");
}
}
else
{
    printf("wrong input\n");
}
}
else
{
    printf("wrong input\n");
}
printf("continues ? (1 for continues) : ");
scanf("%d", &con);
//เป็นการถามว่าจะทำต่อไหม และกด 1 เพื่อทำอีกครั้ง
}
}
```

Result

```
choose an array (1 for 1D, 2 for 2D, 3 for 3D) : 1
input first slot size (not more than 10) : 2
address of a[0] is 1703676
address of a[1] is 1703680
continues ? (1 for continues) : 1
choose an array (1 for 1D, 2 for 2D, 3 for 3D) : 2
input first slot size (not more than 10) : 2
input second slot size (not more than 10) : 2

address of b[0][0] is 1703276    address of b[0][1] is 1703280
address of b[1][0] is 1703284    address of b[1][1] is 1703288

continues ? (1 for continues) : 1
choose an array (1 for 1D, 2 for 2D, 3 for 3D) : 3
input first slot size (not more than 10) : 2
input second slot size (not more than 10) : 2
input third slot size (not more than 10) : 2
in k layer(s) 0
address of c[0][0][0] is 1699276    address of c[0][1][0] is 1699280
address of c[1][0][0] is 1699284    address of c[1][1][0] is 1699288

in k layer(s) 1
address of c[0][0][1] is 1699292    address of c[0][1][1] is 1699296
address of c[1][0][1] is 1699300    address of c[1][1][1] is 1699304

continues ? (1 for continues) : |
```

