```
//35 random and sort ascending or descending function
#include <stdio.h>
#include <stdlib.h>

void ascending();
void descending();
int arr3d[10][10][10], arr1d[30], i, j, k, l, s1, s2, s3, tmp, choice;
//สร้าง arr3d[10][10][10], arr1d[30], i, j, k, l, s1, s2, s3, tmp, choice เป็น global


int main()
{
	int con;
	con = 1;
	while(con == 1)
	{
		l = 0;
		printf("choose sorting type (1 for ascending, 2 for descending) : ");
		scanf("%d", &choice);
		//ให้เลือกว่าจะเรียงจากน้อยไปมาก (ascending) หรือมากไปน้อย (descending)
		printf("input first slot size (not more than 10) : ");
		scanf("%d", &s1);
		printf("input secound 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) && (choice == 1 || choice == 2))
		//ถ้าแต่ละช่องไม่เกิน 10 และ choice เป็น 1 ไม่ก็ 2 ก็จะทำข้างใน
		{
			for(k = 0 ; k < s3 ; k++)
			{
				for(i = 0 ; i < s1 ; i++)
				{
					for(j = 0 ; j < s2 ; j++)
					{
						arr3d[i][j][k] = 0;
						//set default
						arr3d[i][j][k] = rand() % 200 + 1 ;
						//rand() % 200 + 1 คือการสุ่มค่าตั้งแต่ 1 - 200
						//ให้array 3 มิติช่องนั้นๆมีค่าเท่าที่ไปสุ่มมา
						arr1d[l] = arr3d[i][j][k];
						l++;
						//ให้ array 3 มิติช่องนั้นๆ ไปเก็บไว้ใน array 1 มิติ เพื่อเอาไปเรียงในขั้นตอนต่อไป
					}
				}
```

```
}

//print before
printf("before sort\n");
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++)
		{
			printf("Array 3D [%d][%d][%d] = %d \t",i ,j, k, arr3d[i][j][k]);
		}
		printf("\n");
	}
	printf("\n");
}

//แสดงผลว่าก่อนเรียงเป็นแบบไหน

if(choice == 1)
//ถ้าเรียงจากน้อยไปมาก
{
	ascending();
	//เรียกใช้ ascending function
}

if(choice == 2)
//ถ้าเรียงจากมากไปน้อย
{
	descending();
	//เรียกใช้ descending function
}

l = 0;
for(k = 0 ; k < s3 ; k++)
{
	for(i = 0 ; i < s1 ; i++)
	{
		for(j = 0 ; j < s2 ; j++)
		{
			arr3d[i][j][k] = arr1d[l] ;
			l++;
		}
	}
}
//เอา array 1 มิติ ที่เรียงมากใส่ไว้ใน array 3 มิติ
```

```
                        //print after
                        printf("after sort\n");
                        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++)
                                        {
                                                printf("Array 3D [%d][%d][%d] = %d \t",i ,j, k, arr3d[i][j][k]);
                                        }
                                        printf("\n");
                                }
                                printf("\n");
                        }
                        //แสดงผลหลังจากการเรียง
                }

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

void ascending()
{
        for(i = 0 ; i < (s1 * s2 * s3) ; i++)
        {
                for(j = i + 1 ; j < (s1 * s2 * s3) ; j++)
                {
                        if(arr1d[i] > arr1d[j])
                        {
                                tmp = arr1d[i] ;
                                arr1d[i] = arr1d[j] ;
                                arr1d[j] = tmp ;
                                //ถ้าตัวที่ i มีค่ามากกว่าตัวที่ j ก็สลับที่
                        }
                }
        }
}

void descending()
{
```

```
        for(i = 0 ; i < (s1 * s2 * s3) ; i++)
        {
                for(j = i + 1 ; j < (s1 * s2 * s3) ; j++)
                {
                        if(arr1d[i] < arr1d[j])
                        {
                                tmp = arr1d[i] ;
                                arr1d[i] = arr1d[j] ;
                                arr1d[j] = tmp ;
                                //ถ้าตัวที่ i มีค่าน้อยกว่าตัวที่ j ก็สลับที่
                        }
                }
        }
}
```

Result

```
choose sorting type (1 for ascending, 2 for descending) : 1
input first slot size (not more than 10) : 2
input secound slot size (not more than 10) : 2
input third slot size (not more than 10) : 2
before sort
in k layer(s) 0
Array 3D [0][0][0] = 131        Array 3D [0][1][0] = 183
Array 3D [1][0][0] = 91         Array 3D [1][1][0] = 57

in k layer(s) 1
Array 3D [0][0][1] = 118        Array 3D [0][1][1] = 196
Array 3D [1][0][1] = 16         Array 3D [1][1][1] = 149

after sort
in k layer(s) 0
Array 3D [0][0][0] = 16         Array 3D [0][1][0] = 57
Array 3D [1][0][0] = 91         Array 3D [1][1][0] = 118

in k layer(s) 1
Array 3D [0][0][1] = 131        Array 3D [0][1][1] = 149
Array 3D [1][0][1] = 183        Array 3D [1][1][1] = 196

continues ? (1 for continues) : 1
choose sorting type (1 for ascending, 2 for descending) : 1
input first slot size (not more than 10) : 89
input secound slot size (not more than 10) : 89
input third slot size (not more than 10) : 89
wrong input
continues ? (1 for continues) :
```

```
choose sorting type (1 for ascending, 2 for descending) : 2
input first slot size (not more than 10) : 2
input secound slot size (not more than 10) : 2
input third slot size (not more than 10) : 2
before sort
in k layer(s) 0
Array 3D [0][0][0] = 127        Array 3D [0][1][0] = 5
Array 3D [1][0][0] = 159        Array 3D [1][1][0] = 172

in k layer(s) 1
Array 3D [0][0][1] = 80         Array 3D [0][1][1] = 93
Array 3D [1][0][1] = 161        Array 3D [1][1][1] = 13

after sort
in k layer(s) 0
Array 3D [0][0][0] = 172        Array 3D [0][1][0] = 161
Array 3D [1][0][0] = 159        Array 3D [1][1][0] = 127

in k layer(s) 1
Array 3D [0][0][1] = 93         Array 3D [0][1][1] = 80
Array 3D [1][0][1] = 13         Array 3D [1][1][1] = 5

continues ? (1 for continues) : 1
choose sorting type (1 for ascending, 2 for descending) : 2
input first slot size (not more than 10) : 54
input secound slot size (not more than 10) : 6
input third slot size (not more than 10) : 5
wrong input
continues ? (1 for continues) :
```