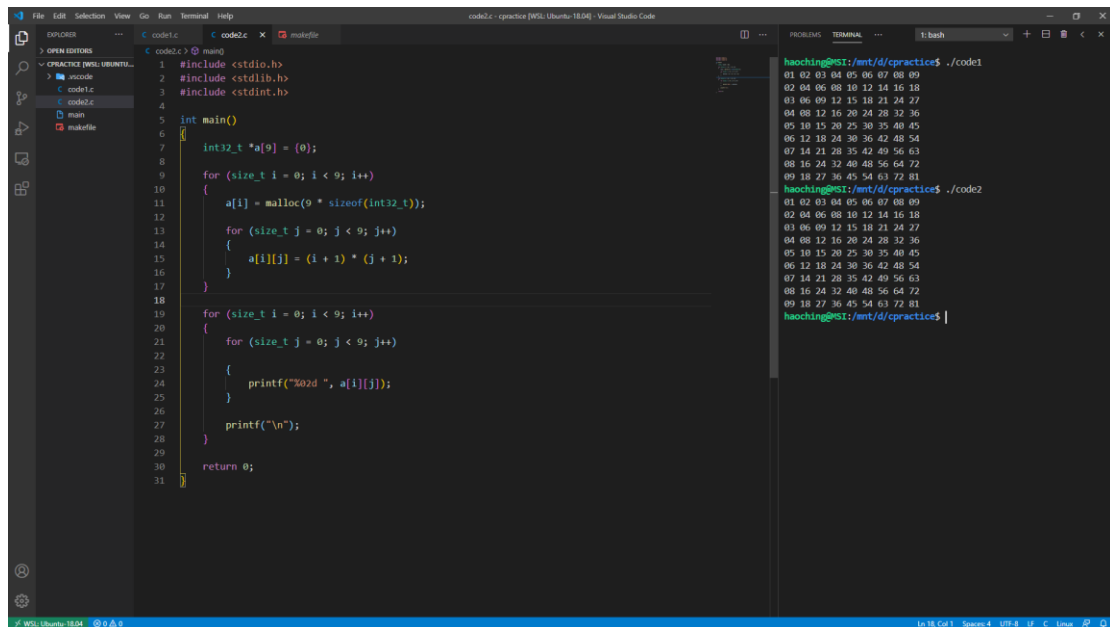


# What is the difference



The screenshot shows a Visual Studio Code editor with two C programs, `code1.c` and `code2.c`, and their execution results in the terminal.

**code1.c:**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4
5 int main()
6 {
7     int32_t *a[9] = {0};
8
9     for (size_t i = 0; i < 9; i++)
10     {
11         a[i] = malloc(9 * sizeof(int32_t));
12         for (size_t j = 0; j < 9; j++)
13         {
14             a[i][j] = (i + 1) * (j + 1);
15         }
16     }
17
18     for (size_t i = 0; i < 9; i++)
19     {
20         for (size_t j = 0; j < 9; j++)
21         {
22             printf("%02d ", a[i][j]);
23         }
24         printf("\n");
25     }
26
27     return 0;
28 }
```

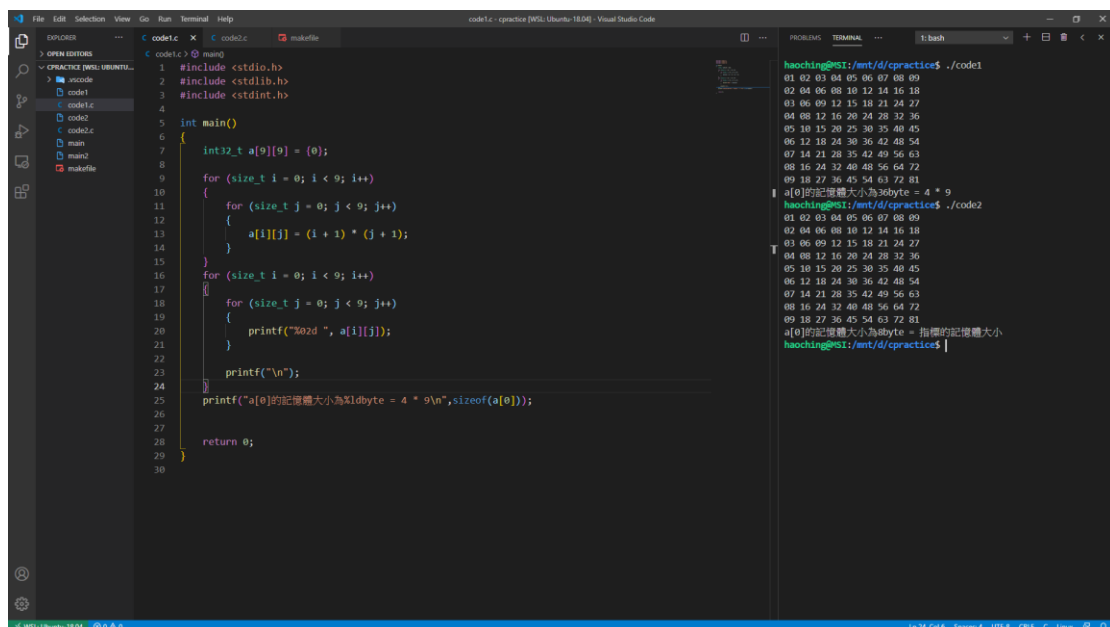
**code2.c:**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4
5 int main()
6 {
7     int32_t a[9][9] = {0};
8
9     for (size_t i = 0; i < 9; i++)
10     {
11         for (size_t j = 0; j < 9; j++)
12         {
13             a[i][j] = (i + 1) * (j + 1);
14         }
15     }
16
17     for (size_t i = 0; i < 9; i++)
18     {
19         for (size_t j = 0; j < 9; j++)
20         {
21             printf("%02d ", a[i][j]);
22         }
23         printf("\n");
24     }
25
26     printf("a[0]的記憶體大小為%ldbyte = 4 * 9\n", sizeof(a[0]));
27
28     return 0;
29 }
```

**Terminal Output:**

```
haoching@51:/mnt/d/cpractice$ ./code1
01 02 03 04 05 06 07 08 09
02 04 06 08 10 12 14 16 18
03 06 09 12 15 18 21 24 27
04 08 12 16 20 24 28 32 36
05 10 15 20 25 30 35 40 45
06 12 18 24 30 36 42 48 54
07 14 21 28 35 42 49 56 63
08 16 24 32 40 48 56 64 72
09 18 27 36 45 54 63 72 81
haoching@51:/mnt/d/cpractice$ ./code2
01 02 03 04 05 06 07 08 09
02 04 06 08 10 12 14 16 18
03 06 09 12 15 18 21 24 27
04 08 12 16 20 24 28 32 36
05 10 15 20 25 30 35 40 45
06 12 18 24 30 36 42 48 54
07 14 21 28 35 42 49 56 63
08 16 24 32 40 48 56 64 72
09 18 27 36 45 54 63 72 81
haoching@51:/mnt/d/cpractice$
```

兩段程式執行結果相同，差別在於 code2 使用動態配置記憶體(先宣告一個 9 元素之指標陣列，再動態配置 9 個記憶體大小為整數的記憶體給每一指標)



The screenshot shows a Visual Studio Code editor with two C programs, `code1.c` and `code2.c`, and their execution results in the terminal.

**code1.c:**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4
5 int main()
6 {
7     int32_t *a[9] = {0};
8
9     for (size_t i = 0; i < 9; i++)
10     {
11         for (size_t j = 0; j < 9; j++)
12         {
13             a[i][j] = (i + 1) * (j + 1);
14         }
15     }
16
17     for (size_t i = 0; i < 9; i++)
18     {
19         for (size_t j = 0; j < 9; j++)
20         {
21             printf("%02d ", a[i][j]);
22         }
23         printf("\n");
24     }
25
26     printf("a[0]的記憶體大小為%ldbyte = 4 * 9\n", sizeof(a[0]));
27
28     return 0;
29 }
```

**code2.c:**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4
5 int main()
6 {
7     int32_t a[9][9] = {0};
8
9     for (size_t i = 0; i < 9; i++)
10     {
11         for (size_t j = 0; j < 9; j++)
12         {
13             a[i][j] = (i + 1) * (j + 1);
14         }
15     }
16
17     for (size_t i = 0; i < 9; i++)
18     {
19         for (size_t j = 0; j < 9; j++)
20         {
21             printf("%02d ", a[i][j]);
22         }
23         printf("\n");
24     }
25
26     printf("a[0]的記憶體大小為%ldbyte = 4 * 9\n", sizeof(a[0]));
27
28     return 0;
29 }
```

**Terminal Output:**

```
haoching@51:/mnt/d/cpractice$ ./code1
01 02 03 04 05 06 07 08 09
02 04 06 08 10 12 14 16 18
03 06 09 12 15 18 21 24 27
04 08 12 16 20 24 28 32 36
05 10 15 20 25 30 35 40 45
06 12 18 24 30 36 42 48 54
07 14 21 28 35 42 49 56 63
08 16 24 32 40 48 56 64 72
09 18 27 36 45 54 63 72 81
haoching@51:/mnt/d/cpractice$ ./code2
01 02 03 04 05 06 07 08 09
02 04 06 08 10 12 14 16 18
03 06 09 12 15 18 21 24 27
04 08 12 16 20 24 28 32 36
05 10 15 20 25 30 35 40 45
06 12 18 24 30 36 42 48 54
07 14 21 28 35 42 49 56 63
08 16 24 32 40 48 56 64 72
09 18 27 36 45 54 63 72 81
haoching@51:/mnt/d/cpractice$
```