

정수 자료형	최소 크기 (Byte)	값의 범위	형식 지정자 Format Specifier
signed char	1	-128 ~ 127	hhd 또는 c (문자)
unsigned char	1	0 ~ 255	hhu 또는 c (문자)
signed short int	2	-32,768 ~ 32,767	hd
unsigned short int	2	0 ~ 65,535	hu
signed int 또는 signed int	2 또는 4	-32,768 ~ 32,767 or -2,147,483,648 ~ 2,147,483,647	d 또는 i
unsigned int	2 또는 4	0 ~ 65,535 or 0 ~ 4,294,967,295	u
long int	4	-2,147,483,648 ~ 2,147,483,647	ld
unsigned long int	4	0 ~ 4,294,967,295	lu
long long int	8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	lld
unsigned long long int	8	0 ~ 18,446,744,073,709,551,615	llu

← → ↺ ↻ 🔍 📄

형식 지정자를 잘 못 사용할 경우 overflow 문제가 생겨서 유도한 데로 출력이 안될 수도 있다.

The screenshot shows a Visual Studio IDE with a C program in the main editor and its output in the Microsoft Visual Studio Debug Console. The program defines several integer types and assigns values to them, including values that cause overflow. The output shows the actual values stored in memory, which differ from the intended values due to overflow.

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4  #include <limits.h>
5  #include <stdlib.h>
6
7  int main()
8  {
9      char c = 65;
10     short s = 200;
11     unsigned int ui = 3000000000U; // 3'000'000'000U
12     long l = 65537L;
13     long long ll = 12345678908642LL; // 12'345'678'908'642LL
14
15     printf("char = %hhd, %d, %c\n", c, c, c);
16     printf("short = %hhd, %hd, %d\n", s, s, s);
17     printf("unsigned int = %u, %d\n", ui, ui);
18     printf("long = %ld, %hd\n", l, l);
19     printf("long long = %lld, %ld\n", ll, ll);
20
21     return 0;
22 }

```

Microsoft Visual Studio Debug Console Output:

```

char = 65, 65, A
short = -56, 200, 200
unsigned int = 3000000000, -1294967296
long = 65537, 1
long long = 12345678908642, 1942899938
D:\github-repository\TBC\Chapter3\Debug\Lecture7.exe (process 2)
Press any key to close this window . . .

```