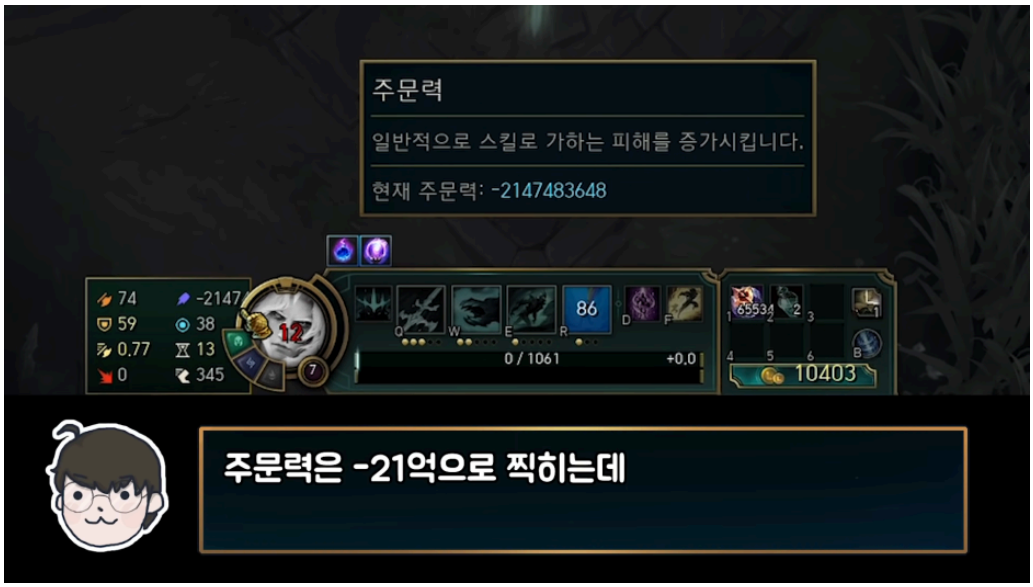
영상을 보니 음수로 표현되는 주문력 값이 실제로는 양수 값으로 처리되는 것 같다.

65534는 unsigned short형이 표현할 수 있는 가장 큰 값인 65535보다 1 작은 값이다.

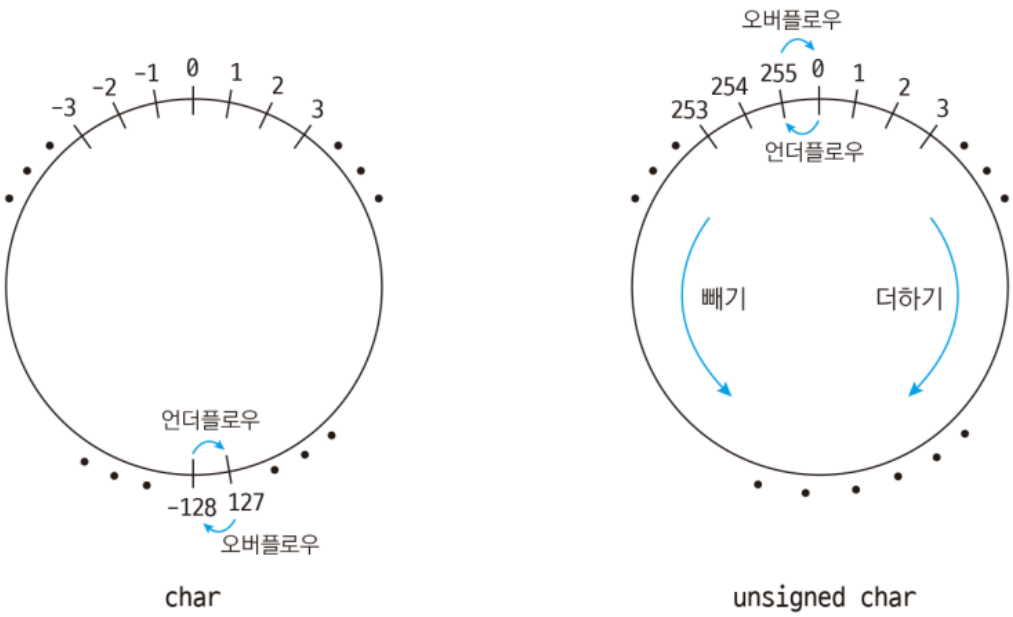


-2147483648은 int형이 표현할 수 있는 가장 작은 값이다.

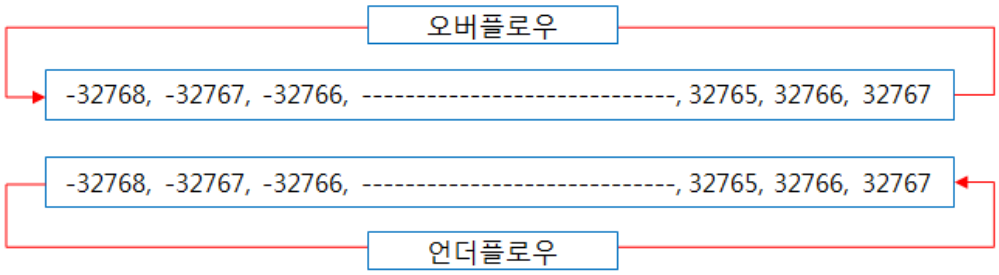
비에고 챔피언 비활성화 버그에서 찾아볼 수 있는 오버플로(overflow) 현상.
2038년 문제로도 잘 알려져 있는 현상이다.
여기서 **오버플로**란 프로그래밍에서 메모리 용량을 넘어선 값이 들어가 생기는 오듀를 말한다.

오버플로 - 나무위키
1 . 사전적 의미 2 . 컴퓨터 용어 2.1 . 10진수 오버플로 2.2 . 8비트 오버…
namu.wiki

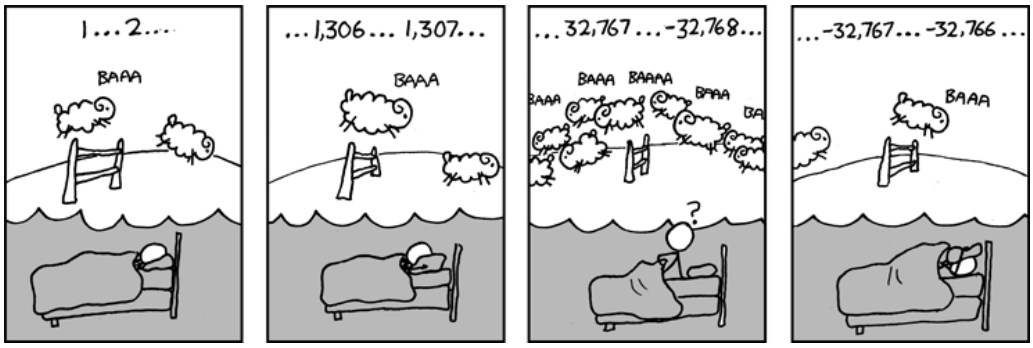
최상위 비트가 부호를 의미할 경우, 해당 자료형 최댓값(최상위 비트 0, 나머지 1)에 1을 더하면 자료형이 표현할 수 있는 최대 범위를 벗어나게 되어 해당 자료형 최솟값(최상위 비트 1, 나머지 0)이 된다. 이를 **정수(integer) 오버플로**라고 한다.
당연히 반대로, 자료형 최솟값에서 1을 빼면 자료형이 표현할 수 있는 최소 범위를 벗어나게 되어 자료형 최댓값이 될 것이다. 이를 **정수 언더플로**(underflow)라고 한다.



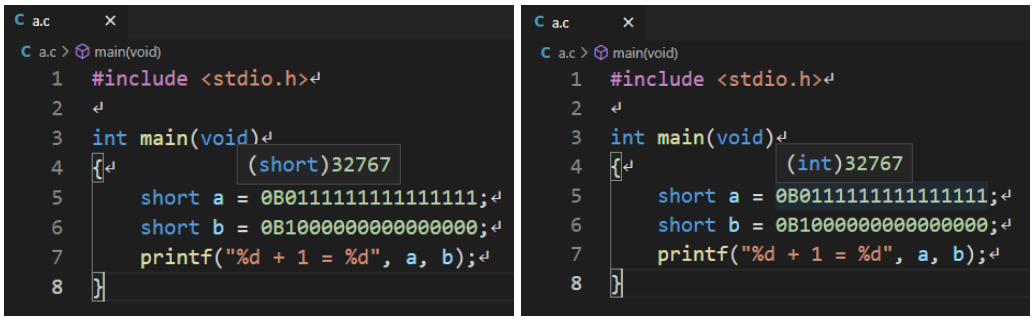
C 언어 코딩 도장: 7.2 오버플로우와 언더플로우 알아보기
◀ 7.1 정수형 변수 선언하기 7.3 자료형 크기 구하기 ▶ 7.2 오버플로우…
dojang.io



여기서 잠깐! 정수 언더플로는 오버플로이므로 일반적으로 언더플로를 뜻하는 산술(arithmetic) 언더플로와 헷갈리지 않도록 하자.
산술 언더플로는 부동소수점 연산에서 지수부가 타입의 한계를 넘어 작아지면 0에 가까워지다가 결국 0이 되어버리는 현상을 의미한다.



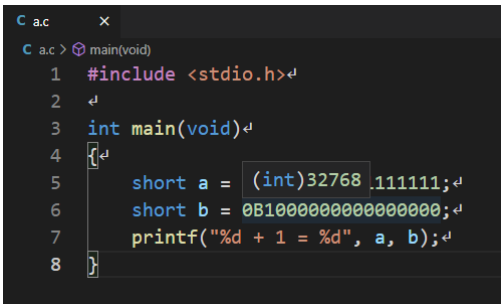
만화에서는 short형으로 정수 오버플로 현상을 설명하고 있다.
숫자가 1씩 증가하다가 3번째 컷에서 short형의 최댓값 32767 다음 급감하여 최솟값 -32768이 되고,
이후 다시 원래대로 1씩 증가하는 것을 확인할 수 있다.
Visual Studio Code에서 C 언어 소스 파일로 직접 확인해보자.



'0B'는 2진법으로 숫자를 표현할 때 대입하는 숫자 앞에 붙이는 것이다.
a는 short형, int형 모두 32767이다.

```c
#include <stdio.h>

int main(void)
{
    short a = (short)(-32768) - 1;
    short b = 0B1000000000000000;
    printf("%d + 1 = %d", a, b);
}
```



```c
#include <stdio.h>

int main(void)
{
    short a = (int)32768 - 111111;
    short b = 0B1000000000000000;
    printf("%d + 1 = %d", a, b);
}
```

a에 1을 더하면 최상위 비트가 1이 되면서
b는 short형으로 나타내면 -32768, short형보다 범위가 넓은 int형에서는 32768이 된다.
여기에 1을 더한 값(0B1000000000000001)은 short형으로는 -32767, int형으로는 32769가 될 것이다.

```
$ ./a.exe
32767 + 1 = -32768
```

프로그램을 실행하면 요래 나온다.
b 값에 '0B1000000000000000' 대신 'a + 1'을 대입해도 동일한 결과가 나온다.
다만 printf 함수 내에서 b 대신 'a + 1'을 넣으면 32768이 나온다.

+
동명의 성인 애니메이션이 있어서 처음 검색해서 공부할 때 굉장히 당황했다 -.-;