

Basic_Concepts_2 - Undefined Behaviour

음수 값이 양수값 보다 표현할 수 있는 범위가 더 크다.

```
int x = std::numeric_limits<int>::max() * -1;
cout << x; // 문제없음

int y = std::numeric_limits<int>::min() * -1;
cout << y; // 오버플로우
```

정수값의 최대 크기보다 큰 값으로 초기화 하는것

```
int z = 3000000000;
```

부호가 있는 정수형 타입에 비트연산을 하는 것

```
int y = 1 << 12;
```

데이터 타입의 비트수보다 큰 시프트 연산

```
unsigned y = 1u << 32u; // unsigned 타입도 예외는 없다.
```

암묵적 형변환으로 범위 밖의 값을 표현하려는 경우

```
uint16_t a = 65535;
uint16_t b = 65535;
cout << (a * b); // '-131071'출력. int타입 오버플로우
```

오버플로우 / 언더플로우

unsigned 타입의 오버플로우/언더플로우 감지는 쉽지 않다. signed 타입의 감지는 더욱 어렵기 때문에, 사용하기 전에 꼭 확인해야 한다.