190418

**\*Break, Continue문**

Break의 역할 : 현재 Block에만 빠져나간다. (Nested Blcok의 경우 break가 속해 있는 곳에서 바로 상위의 Block으로 나간다. )

Continue의 역할 : 현재 속해 있는 block에서 가장 가까운 반복문의 조건부로 이동한다.(즉, Nested Block으로 되어 있다고 해도 그 블록을 빠져나오는게 아니라 가장 가까운 반복문의 조건으로 이동한다.)

※자세한 내용은 비주얼 스튜디오 예제 참고.

**\*난수 만들기**

<cstdlib>\std::rand(), std::srand()

Srand([인자]) : seed-random의 약자로 시작하는 숫자를 생성할 때 인자를 사용하여 생성한다.

Rand() : 랜덤으로 숫자를 돌리는 라이브러리 함수

문제점 ) srand의 인자(seed number)가 고정되어 있는 한 다른 랜덤 숫자를 생성할 수 없다.(첫번째 돌린 랜덤 숫자 결과와 두번째 돌린 랜덤 숫자의 결과가 같게 나온다.)

해결 ) <ctime>\std::time()을 이용하여 해결한다.

**\*ctime 라이브러리**

<ctime>\std::time() : CPU 클락에서 현재 시간을 가져온다.

Tip) 디버깅 할 때는 seed값을 고정해 주는게 좋다. 프로그램에서 문제라고 생각 됐던 현상이 발생했다 안 했다 할 수 있기 때문이다.

**\*random 라이브러리**

<random>\ std::random\_device, std::mt19937, std::mt19937\_64 라는 사용자 정의 자료형과, std::uniform\_int\_distribution<> dice(최소값, 최대값) 라는 사용자 정의 함수를 제공한다.

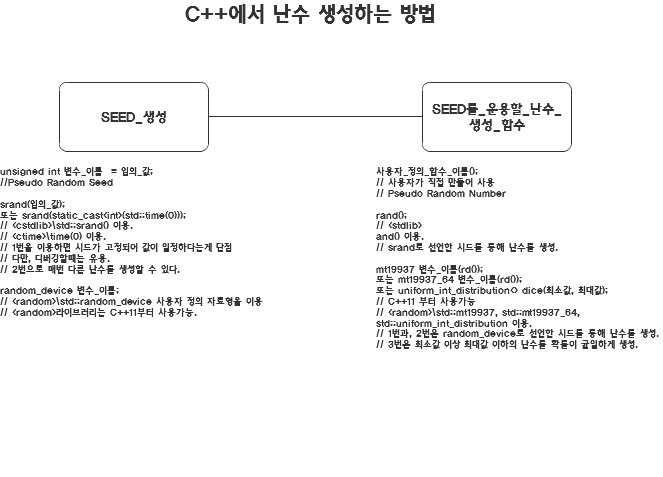
random\_device : ‘rd()’라고 써서 운용하며, 밑에 난수 생성 초기화 시 초기화 값으로 쓰면 좋다.

mt19937 : (seed값을 초기화하여) 32비트짜리 난수를 생성한다.

mt19937\_64 : (seed값을 초기화하여) 64비트짜리 난수를 생성한다.

uniform\_int\_distribution<> 변수명(최소값, 최대값) : 최소값 이상 최대값 이하의 난수를 생성하는데 범위안에 있는 값이 나올 확률이 모두 균일하게 생성한다.

<정리>



\*std::cin 더 잘 쓰기

의의) cin으로 사용자에게 입력을 요구하고, 사용자가 프로그래머가 의도하지 않은 input을 넣었을 때를 대비해 프로그래머가 할 수 있는 일들.

std::cin.ignore(적당히\_큰\_숫자, ‘\n’);

//입력 받은 버퍼를 비워주는 역할.

//보통, 적당히 큰 숫자로 numeric\_limits<std:streamsize>::max()를 넣어준다.

std::cin.fail();

//입력 버퍼에게 물어 입력에 실패했는지 묻는 함수. 실패했을 경우 true를, 아니면 false를 반환한다.