

 **START**

GL

표준 C 라이브러리의 rand() 함수는 [0,RAND_MAX — 1]에서 균
다. rand() mod n이 [0, n -1]에서 균일하게 분포된 숫자를 생성한다고 생각할 수 있
요?



아니요, 그렇지 않습니다. rand() % n을 사용하여 [0, n-1] 범위에서 균일하게 분포된 숫
자를 생성하는 것은 일반적으로 권장되지 않습니다. 이는 rand()가 반환하는 값이
RAND_MAX보다 큰 경우, 그리고 n이 RAND_MAX보다 큰 경우에 결과값이 편향될 가능
성이 있기 때문입니다.

예를 들어, 만약 RAND_MAX가 32767이고 n이 100인 경우, rand() % 100의 결과값은 0
에서 99까지의 숫자가 균일하게 생성되지 않을 수 있습니다. 만약 rand()이 32750을 반
환한다면, 50~99 중의 숫자가 반환될 가능성이 높아집니다.

따라서, 더 나은 대안으로는 rand()를 사용하여 0에서 RAND_MAX까지의 난수를 생성하
고, 이 값을 [0, n-1] 범위로 매핑하는 방법을 사용할 수 있습니다. 예를 들어, 다음과 같
은 코드를 사용할 수 있습니다.

perl

Copy code

```
int r = rand();  
int num = r % n;
```

위 코드 대신에 아래와 같은 코드를 사용하여 더 나은 분포를 얻을 수 있습니다.

arduino

Copy code

```
int r = rand();  
int num = (int)((double)r / ((double)RAND_MAX + 1) * n);
```

위 코드는 0에서 RAND_MAX 사이의 난수 r을 생성한 후, 이를 0에서 n-1까지의 숫자로
매핑합니다. 이렇게 함으로써 균일한 분포를 더 잘 보장할 수 있습니다.



Send a message...



[ChatGPT Mar 23 Version](#). Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts